AMSAA

TECHNICAL MEMORANDUM NO. 151

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS), A USERS' MANUAL

WILBERT J. BROOKS
WARD V. FOSTER
RICHARD T. MARUYAMA

NOVEMBER 1972

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY
Aberdeen Proving Ground, Maryland

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

TECHNICAL MEMORANDUM NO. 151

MATHNET AND RISCA

(NETWORK ANALYZER PROGRAMS)
A USERS' MANUAL

Wilbert J. Brooks
Ward V. Foster
Richard T. Maruyama

November 1972

RDT&E 1P765801MM1102

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY

ABERDEEN PROVING GROUND, MARYLAND

U.S. ARMY MATERIEL SYSTEMS ANALYSIS AGENCY

TECHNICAL MEMORANDUM NO. 151

WJBrooks/WVFoster/RTMaruyama/sm
Aberdeen Proving Ground, Maryland
November 1972

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL

ABSTRACT

Two network analyzer programs, MATHNET and RISCA, which allow the analyst to simulate a general class of network representations are described and evaluated for the potential user.

Network concepts, program listings, and program flow charts are included for both programs in addition to detailed description of input preparation and output interpretation for a hypothetical example.

Next page is blank.

CONTENTS

Next page is blank.

1. INTRODUCTION

1.1 Background.

Although having much wider application, the primary use of network analysis has been in the planning and control of R&D projects. The use of network analysis techniques for this purpose had its origin with PERT (Program Evaluation and Review Technique) in 1958 in the Polaris Program. Since that time, PERT and many other network analysis techniques have gained wide acceptance both in the Department of Defense and in private industry.

In the last two years, two network analyzer programs, MATHNET* and RISCA**, have gained acceptance within AMC. Under PROMAP 70 the responsibility for instruction in risk analysis was assigned to the Army Logistics Management Center (ALMC) at Fort Lee, Virginia. In order to expedite the program, a contract was let to MATHEMATICA by the Army Research Office to develop this course of instruction. MATHNET, developed by MATHEMATICA as a teaching aid for this course was modified by ALMC. They call their version of the program RISCA.

The version of MATHNET currently being used at this agency is not identical to that developed by Mathematica. The original Mathematica program contained several logic inconsistencies with respect to the time and cost values generated in various types of nodes. These logic errors, corrected by analysts at Picatinny Arsenal, are not in the version of MATHNET discussed in this manual. The RISCA version discussed herein is also free of these logic inconsistencies.

Even though the two programs are accepted and the results used, there does not exist an adequately documented users' manual which compares both programs at this time. Recognizing the utility of network analysis techniques in the Materiel Acquisition Decision Making Process and the need for such a manual both within AMSAA and AMC, a project was initiated to prepare this users' manual.

1.2 Organization of the Report.

The remainder of the first section is devoted to defining the various characteristics of a network and the types of networks that can be modeled using RISCA and MATHNET. In Section 2, the various capabilities of the two network analyzer programs are described. In Section 3, a description of the construction of a network and the method for inputting data to the program is provided. The output of

---

* Mathematical Network Analyzer.

** Risk Information System and Cost Analysis.

both programs is described in Section 4. In the last section of the manual, the program capabilities are compared and contrasted and a recommendation is made concerning which program to use.

1.3 Network Concepts.

Before the two network analyzer programs are described and compared, the more basic concepts of a graph, a node, an arc, a network and a path must be defined.

Figure 1.1 is an example of a graph. The circles represent nodes, and the lines joining the nodes are called arcs. Hence, a graph is a collection of two or more nodes joined by arcs. Any arc can be characterized by the pair of nodes that it connects. For example (1,2) characterizes the arc connecting nodes 1 and 2 in Figure 1.1.

The only difference between a graph and a network is that the arcs have some type of flow in them (see Figure 1.2). One example of a system that can be represented by a network is a development test-program. The nodes* in a development test-program represent the initiation or completion of various tests, the arcs** represent the actual tests being conducted and the flow in the arcs is time and/or cost involved in testing.

Finally, a path is defined as a sequence of arcs connecting two nodes. For example, the following sequence of arcs form paths between nodes 1 and 4 in Figure 1.2:

PATH (M) (1,2), (2,4); PATH (0) (1,4)
PATH (N) (1,3), (3,4)

1.4 Types of Network Representations.

Given the preceding concepts, it is now possible to describe the different types of network representations and network analysis techniques for analyzing them. The differences in the networks result from assumptions made concerning the events and the flows in the activities being modeled in the project. As mentioned previously, the arcs represent activites, the nodes represent events, and the flow in the arcs usually represents time and/or cost.

---

*Nodes generally refer to events.

**Arcs generally refer to activities or jobs.

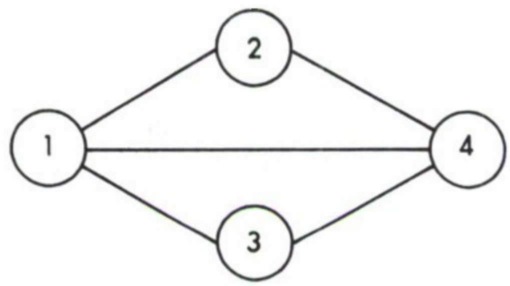


Figure 1.1 Example of a Graph.

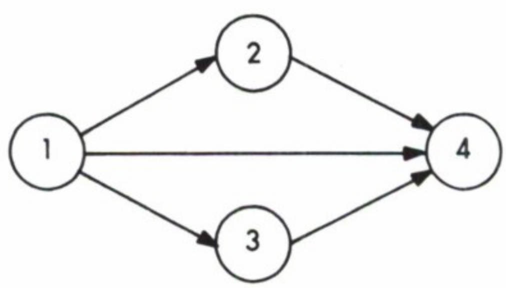


Figure 1.2 Example of a Network.

Three types of network models will be discussed. They are:

a. Deterministic event and activity time/cost network

b. Deterministic event and probabilistic activity time/cost network

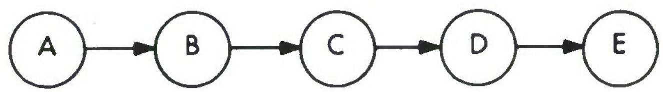
c. Probabilistic event and activity time/cost network

It should be pointed out that this discussion will be very general and only the major attributes and assumptions about the networks being modeled will be discussed.

The differences in the types of network representations are most easily described by analyzing one example under the varying assumptions made concerning the events and flows in the activities. Changing the oil in a car is the example that will be used throughout this discussion.

The first type of possible network representation for describing the events and activities involved in changing the oil in a car is illustrated in Figure 1.3. Here there are five milestone events in changing the oil in a car. For this network representation, as well as all others, it is assumed that the events must be completed in a particular sequence in order to complete the project. Further, it is assumed that all events must be completed and the completion times are known with certainty (i.e., the events and completion times are assumed to be deterministic). For most programs, these last two assumptions are not thought to be very realistic. For rarely are the events and/or activity times known with certainty. Even though the first type of network representation is not realistic for R&D projects, it is realistic in the construction industry where tasks for a project are known with certainty. Further, these tasks are repetitive so the assumption of deterministic activity times is more realistic. The Critical Path Methodology (CPM) is the name given to the network analysis technique developed by DuPont in 1958 to handle deterministic event and activity time networks (Type I). This technique was initially used to find an efficient method for planning the construction of a new facility.

In the second network type, the assumption of deterministic activity times is replaced by the assumption of probabilistic activity times. This means that the activity times are not known with certainty (i.e., there exists some distribution of activity times). For instance, the activity time for BC in the changing the oil example could vary due to random interruptions such as having to pump gas for cars as they arrive at the station. For instance, the most likely time for completing this activity might be ten minutes and the best and worst times might be five and twenty minutes respectively. Therefore, if this activity time is assumed to be distributed as a triangular distribution, with a minimum, most likely and maximum



EVENTS

A – CAR IS LEFT AT THE SERVICE STATION.

B – CAR IS ON THE LIFT.

C – OLD OIL IS DRAINED AND THE FILTER REMOVED.

D – NEW FILTER AND OIL IN THE CAR.

E – CAR OFF THE LIFT.

ACTIVITY TIMES

ARC

AB – 2 MINUTES

BC – 5 MINUTES

CD – 6 MINUTES

DE – 2 MINUTES

Figure 1.3. Changing the Oil in a Car.

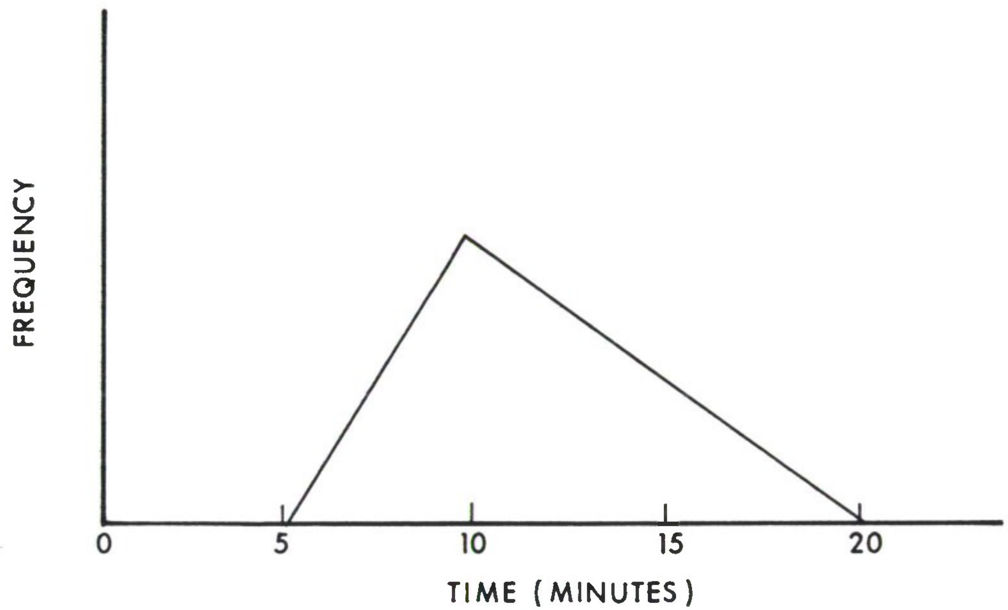
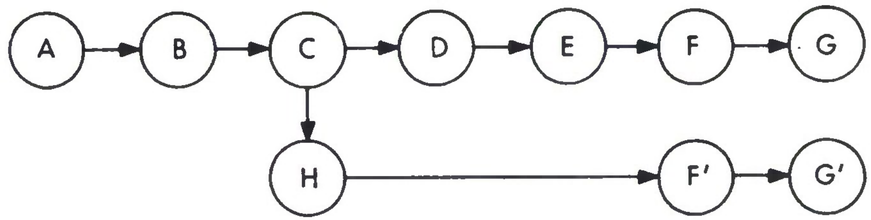


Figure 1.4 Distribution of Activity Time BC.



EVENTS

A – CAR IS LEFT AT THE SERVICE STATION.
B – CAR IS ON THE LIFT.
C – OIL IS DRAINED.
D – FILTER IS REMOVED.
E – NEW FILTER IS IN THE CAR.
F' AND F – NEW OIL IS IN THE CAR.
G' AND G – CAR OFF THE LIFT.
H – FILTER IS NOT REMOVED BECAUSE IT CANNOT BE REPLACED.

Figure 1.5. Changing the Oil in a Car.
(Probabilistic Network)

activity times being 5, 10, and 20 minutes respectively, then Figure 1.4 represents the distribution of this activity time. The network analysis technique developed to analyze this type of network representation is PERT (Program Evaluation and Review Technique) which was developed in 1958 for planning and controlling the development of the Polaris missile. Even though PERT is more realistic than the CPM, it still is deficient for modeling many R&D projects since the events are assumed to be deterministic.

In the third network type, all events and activities are modeled probabilistically. This type is the most realistic for modeling R&D programs. To illustrate this network type, assume that it may not always be possible, for example, to replace the oil filter because the type of filter required may not be in stock. Assume that there is a .98 probability that it is in stock and a .02 probability that the filter is not in stock. The probabilistic nature of this network type is illustrated in Figure 1.5. In addition, when each of the activity times is modeled probabilistically, this network type is beyond the scope of PERT and CPM and due to the additional complexity introduced by probabilistic events, the development of RISCA and MATHNET was prompted.

It should be noted that both MATHNET and RISCA also allow one to analyze the first two network types.

2. RISCA AND MATHNET CHARACTERISTICS AND CAPABILITIES

RISCA and MATHNET are computer programs that allow one to analyze systems that can be represented by a general class of networks. Since the events and activity times and/or costs can be modeled probabilistically, a simulation process is utilized. The output consists of a frequency of occurrence distribution for each of all possible terminal events and corresponding time and/or cost distribution for each terminal event. In addition, the distribution of time and/or cost weighted over all possible terminal events is estimated.

Many of the benefits derived from analyzing a network result from the analysis and thinking that is required in the construction of the network. Consider, for example, the development of a tank where there are several alternative designs. Describing the sequence of events for alternate development programs for each design provides insight into the types of problems that one is likely to encounter in each program.

The oil changing example (Figure 1.5) from the introduction is utilized to demonstrate the characteristics and capabilities of MATHNET and RISCA.

Since in this example the event, "removing the oil filter," is the only uncertain event, the probabilistic event network can be described in terms of two deterministic-event sub-networks. One sub-network represents the events and activities involved in changing the oil and the oil filter and the other represents the events and activities involved in changing only the oil. In this simple example there is only one path in each sub-network; however, in more realistic problems there will almost certainly be several possible paths in a sub-network.

Monte Carlo procedures are used to determine which deterministic event sub-network will be followed in the probabilistic event network. Each of the sub-networks have a terminal event whose completion time is determined by Monte Carloing all the activity completion time distributions within the sub-network. All of the potential paths in the sub-network are then investigated using these sample activity times. If the sub-network chosen in the oil changing example involves removal of the oil filter, then the activity time distributions in this sub-network would be randomly sampled. These sample values would then be summed to estimate the sub-network completion time. In addition, the cost of all activities in the sub-network would be sampled and summed to estimate completion costs. However, in this example costs were not considered.

The preceding procedure is repeated many times, and the sampling distributions of terminal events and time and/or cost are constructed. It should be pointed out that in reality the deterministic event sub-network is chosen and the corresponding time and cost estimates for each activity on this network are accumulated as the network is simulated.

For this example, assume that 300 iterations have been run. Both RISCA and MATHNET would provide the frequency histograms of the percentage of times each terminal event was selected as shown in Figure 2.1 and the completion time distributions shown in Figures 2.2, 2.3, and 2.4. In addition, RISCA would provide a cumulative distribution of time for each completion time distribution in Figures 2.2, 2.3, and 2.4. Further discussion, interpretation and comparison of the two programs' output are deferred to the Output Section.

If cost is considered, there are two options available: (1) the cost can be estimated independent of time by running a separate simulation or (2) the cost can be estimated as a linear function of time in the same simulation, i.e., cost = (fixed cost)+(variable cost)x (time).

In addition to the insight derived in structuring and simulating the network, this type of analysis can provide a framework for evaluating and consolidating relevant information for decision making purposes. If one is trying to select an alternative system, this type

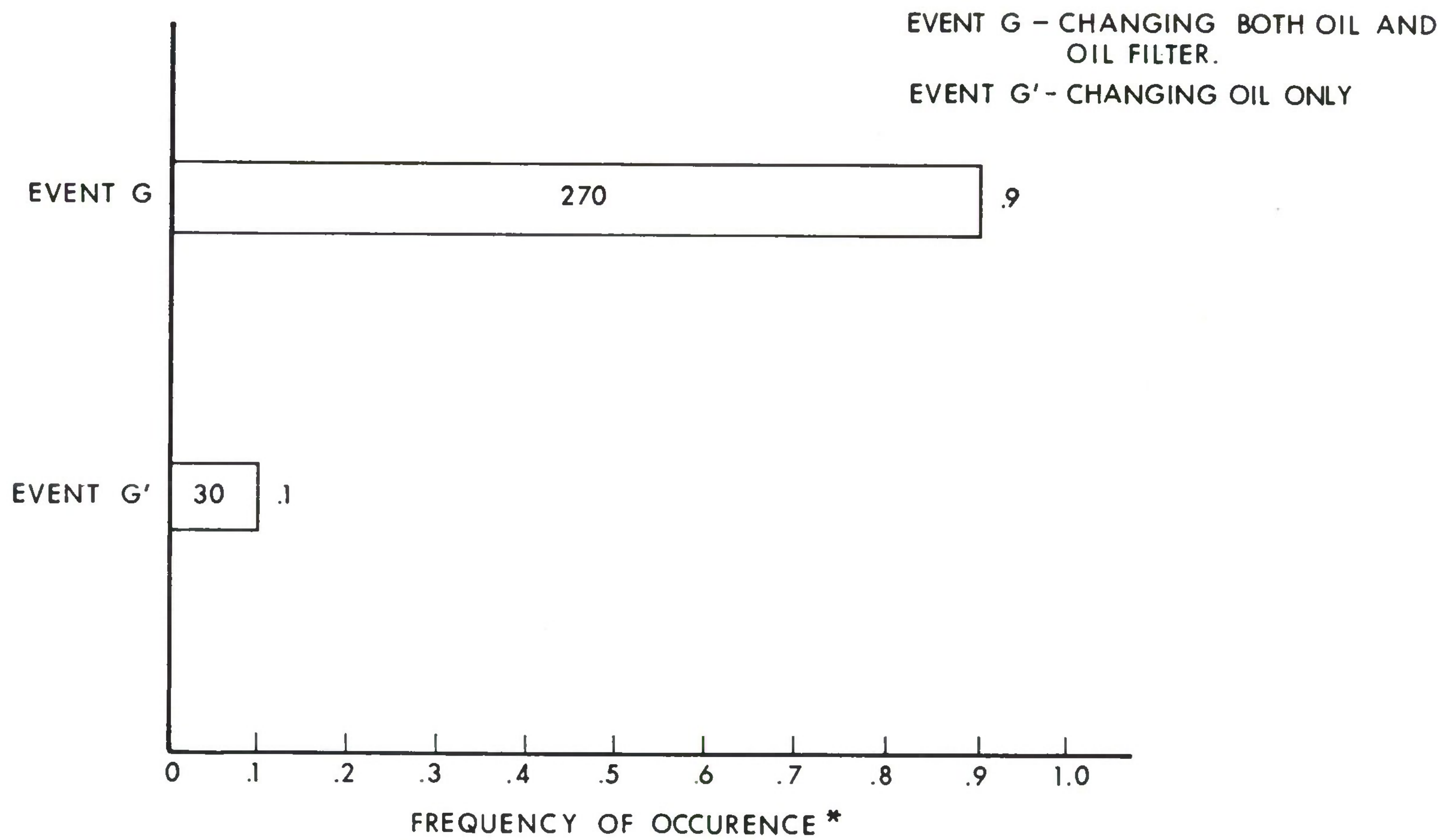


Figure 2.1 Possible Terminal Events.

* PERCENT OF TIMES THE TERMINAL EVENT WAS SELECTED.

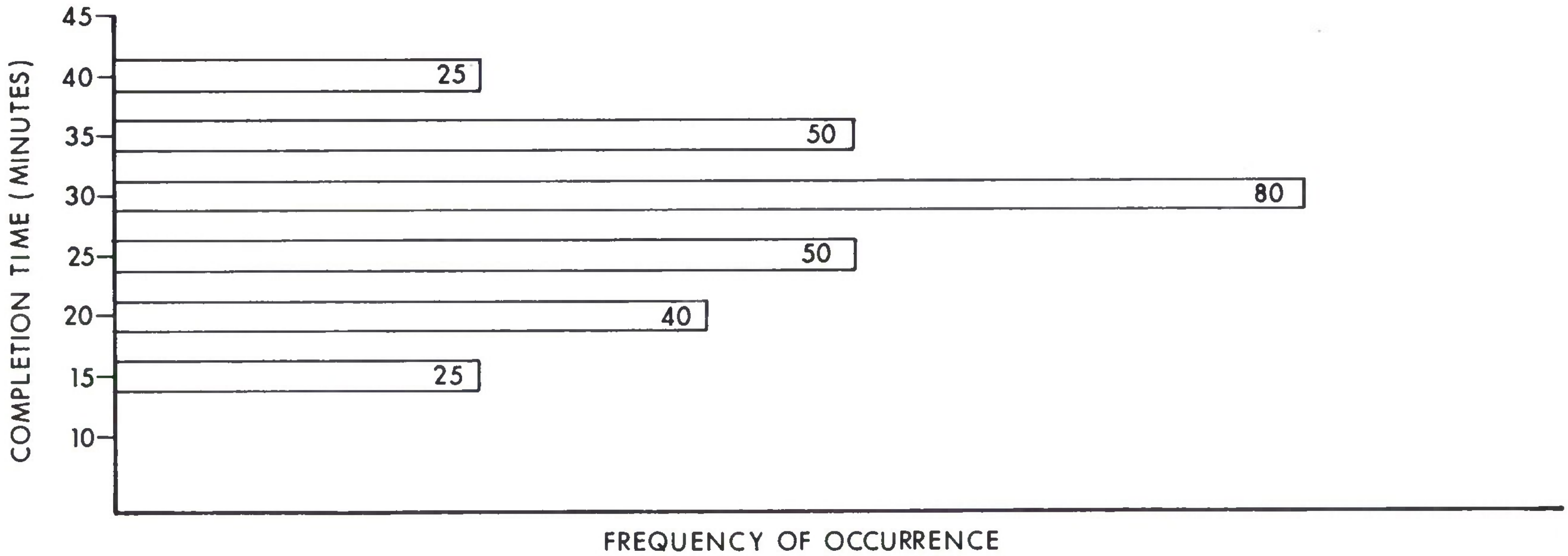


Figure 2.2 Frequency Histogram of Completion Times for Terminal Event G.

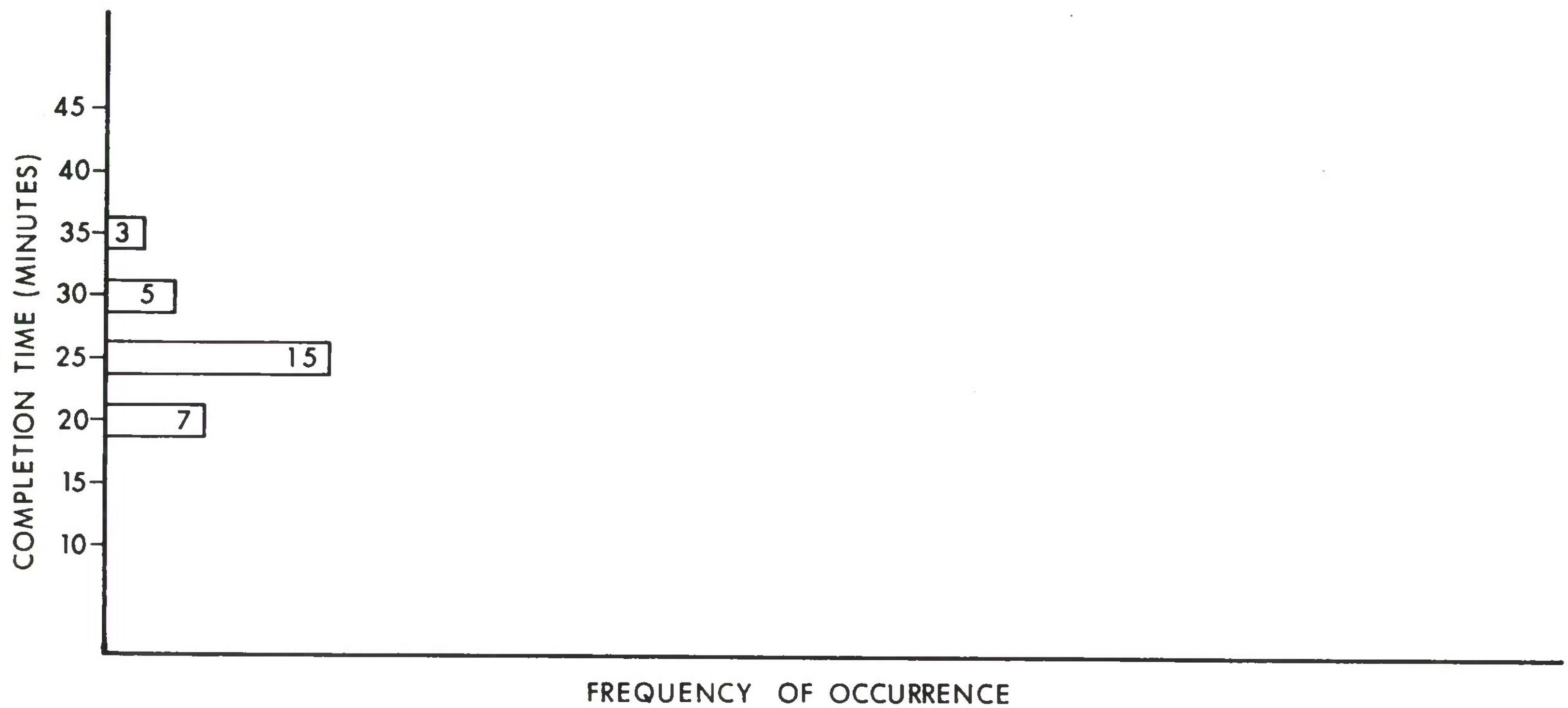


Figure 2.3 Frequency Histogram of Completion Times for Terminal Event G′

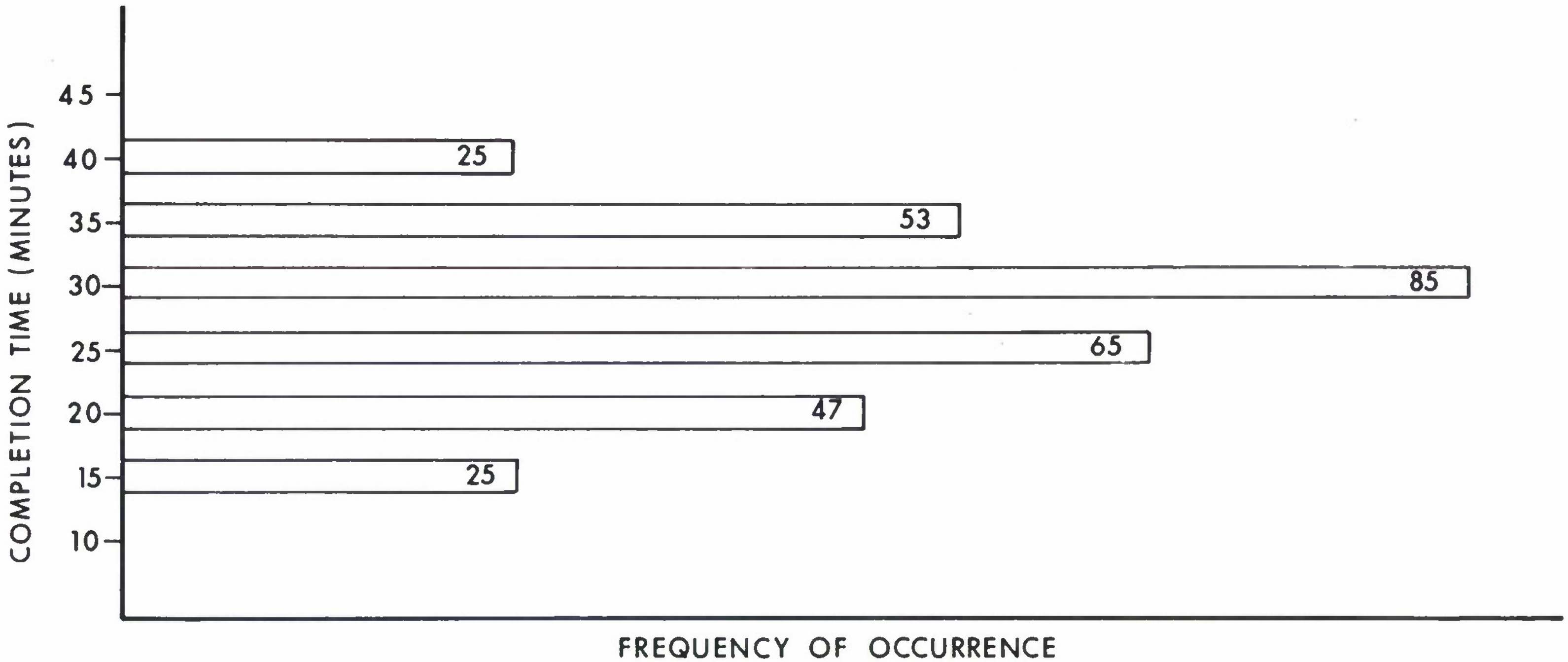


Figure 2.4 Frequency Histogram for Project Completion Times Weighted Over Terminal Events G and G′.

of analysis can provide a framework for evaluating and consolidating relevant information for decision making purposes. If one is trying to select an alternative system, this type of analysis can provide a reasonable framework for comparing time and cost trade-offs in light of possible terminal events for each possible alternative. Consider, for example, the development of a tank where there are alternate designs. These alternate designs could be compared on the chances of program failure (i.e., having to cancel the development) and the chances of developing the tank for a fixed number of dollars in a fixed period of time. In addition, if the chances of failure are great with a preferred alternative, then this analysis might serve as the basis for restructuring the development program to allow for parallel development of two designs thereby increasing the chances of successful development.

In addition, this type of analysis could provide a framework for examining the progress in a program and provide inputs to determine whether or not to continue the program or to assess the impact of program changes. Once the network has been constructed, it is a relatively simple matter to modify the basic structure or update the data. Therefore, this technique could provide an up-to-date statement of the chances of meeting program time and cost objectives.

Using these programs is no substitute for analysis. Major effort still must be applied in trying to realistically model the system as a network. These programs are only tools. If a conscientious job of modeling the system has been accomplished, then the output of these programs is useful for both planning and decision making. The uses of the output in a decision making context will be discussed in more detail in the Output Section.

3. RISCA AND MATHNET PROGRAM INPUT

3.1 Introduction.

This section describes how the elements of a network are coded and inputted into RISCA and MATHNET.

Included is a description of the arc and node notation, the program input formats, an example problem and a comparison of the two input decks used in the example.

It is important to note that all of the arc and node notation and the input formats described herein are the same for both RISCA and MATHNET. MATHNET, for time sharing mode, contains several additional control cards. The nature of these cards and their use will be fully explained when the input decks for the example are compared.

3.2 Arc and Node Notation.

As explained in the introduction, the two classes of symbols used in describing a network are arcs and nodes. An arc is used to connect two nodes and represents an activity. A node is used to represent a decision or the initiation or termination of an activity.

Each arc inputted to the RISCA* network analyzer program is characterized by:

a. An arc name

b. The name of the initial node

c. The name of the terminal node

d. A distribution of completion times

e. A distribution of activity costs (as a linear function of time)

f. The probability of successful completion

The following time distribution types are available:

a. Normal

b. Triangular

c. Uniform

d. Constant

Arcs may exist in one of the following states:

a. Idle - the activity associated with the arc has not been initiated

b. Initiated - the activity associated with the arc has been initiated, but its outcome has not been determined

c. Completed - the activity associated with the arc has been successfully completed

d. Unsuccessful - the activity associated with the arc has been initiated but not successfully completed

---

*This also applies to MATHNET

e. Utilized - the node which terminates the arc has been satisfied.

The conditions which determine in which of the above states an arc will be are discussed following a description of the node types.

Nodes represent major events or decisions. Each node inputted to the program is characterized by:

a. a node name

b. an input rule

c. an output rule

In addition, time and/or cost values are also assigned to the nodes. However, these values vary (except for constant distributions) with each iteration. The time and/or cost values assigned to the nodes are those derived from the arcs entering them. In the cases where more than one arc enters a node, the exact values assigned to the node are dependent upon that node's input rule. These rules are described in detail in this section.

In order for an activity to be completed, it must satisfy the conditions of the node that it enters. For example, assume that an <u>activity</u> is "developing a model" and the terminating <u>event</u> is "completion of the model". If the model is completed, the arc is said to be <u>successful</u> and the node "satisfied".

The rules which determine whether or not the nodes of the network are satisfied are the input rules. They are:

a. <u>And</u> Input - All input arcs must be successfully completed. The time value assigned to the node is the maximum of all input arcs. The cost value is the sum of the costs generated by each input arc.

b. <u>Or</u> Input - At least one input arc must be successfully completed. The time value assigned is the minimum of all input arcs successfully completed. The cost value is the sum of the costs generated by each input arc <u>based</u> upon the smallest time value.

When an activity has satisfied the input rule of a node, the nature of its exit from the node is dependent upon the node output rules. There are two such rules:

a. All - all output arcs are initiated simultaneously.

b. Probabilistic - Associated with each output arc is a probability. The sum of the probabilities exiting one node must equal one.

One important point should be stressed at this time. The probabilities assigned by the analyst to each arc emanating from a node are done so on node data cards. This probability that an arc will be initiated is not to be confused with the probability that an arc will be successfully completed. This latter information is entered on an arc data card.

There are five special nodes which do not conform to the input and output rules outlined above. They are:

a. Initial Node - used to initiate the simulation. It possesses no input arcs

b. Terminal Node - used to terminate simulation. It possesses no output arcs

c. $\overline{1/1}$ (one-to-one bar) Node - This node possesses N input arcs and N+1 output arcs. Associated with each input arc is a unique output arc. The extra output arc is a default arc. The time value assigned to this node is the maximum of the successfully completed input arcs. The cost is the sum of the cost values for each of the input arcs which are completed successfully. This time and cost assignment is true for the remaining node types in this discussion. This node is satisfied in one of two ways:

(1) by having one successful input arc (all input arcs must be initiated) which will result in one output arc being initiated

(2) by initiation of the default arc. This is done when all of the input arcs have been initiated but none successfully completed.

The one-to-one bar node is used to simulate the time, cost and probability of completion for competing activities. For example, if two contractors submit the time, costs and probabilities associated with producing a certain prototype, this data can be applied to two arcs entering a $\overline{1/1}$ Node. By examining the corresponding output arcs and the default arc, the developer can gain valuable insight from success and failure probabilities, as well as the time and cost involved with each alternative.

d. 1/1 (one-to-one) Node - This node possesses N input arcs. Associated with each input arc is a unique output arc. The node will be satisfied by one complete input arc. One output arc will be initiated, namely the one associated with the input arc satisfying the node. 1/1 Nodes are used to preserve the time and cost factors for two or more activities experiencing a common event (the event being represented by the 1/1 Node). For example, suppose two contractors are developing blueprints for a house and the cost and time for each contractor differs. If the probability that both will

be successfully completed is 1.0, each of these activities (arcs) may enter a common node (blueprint completion). As the arcs exit the node, the time and cost of each activity is preserved.

Unlike the $\overline{\text{I/I}}$ node, there is no default arc to be initiated when activities are unsuccessful. Therefore all arcs entering the I/I node must have a probability of completion 1.0. A probability of less than 1.0 may cause a premature termination of the simulation resulting in no output.

e. Preferred Node - This node possesses N input arcs and N+1 output arcs. Associated with each input arc is a unique output arc. The extra output arc is a default arc. The preferred node differs from a $\overline{\text{I/I}}$ node in that the input/output arc pairs are ordered by the users preference. This node is satisfied in two ways:

(1) If all the input arcs are initiated and at least one arc has been successfully completed the output arc associated with the most preferred of the complete input arcs will be initiated.

(2) If all the input arcs are in the unsuccessful state the default output is initiated.

It is important to note that this node will never be satisfied unless all input arcs have been initiated and are in either the complete or unsuccessful state.

This node is used to simulate the time, cost and probability of completion for competing activities when the activities can be listed by order of preference.

A standard set of symbols is used to identify the various node types and input/output rules in a schematic network representation. A listing of these node types and the symbols used to represent them is found in Table 3.1.

3.3 Input Format.

The format used to enter the arc and node data into the RISCA network analyzer program will be discussed.

The Data deck consists of 3 blocks:

Block I - Header Card

Block II - ARC Cards

Block III - Node Cards

TABLE 3.1 SYMBOLIC REPRESENTATION OF INPUT/OUTPUT RULES AND SPECIAL NODES

INPUT RULES	SYMBOL
AND	AND
OR	OR
INITIAL	INIT
OUTPUT RULES	
ALL	ALL
PROB	PROB
TERMINAL	TERM
SPECIAL NODES	
ONE-TO-ONE	1/1
ONE-TO-ONE BAR	$\overline{1/1}$
PREFERRED	PREF

A description of each follows:

Block I: The header card is one card used to describe the network being analyzed. All 80 columns may be used.

Block II: The information required for each arc appears on one card which is formatted as follows:

	Info	Columns	Format
a.	Arc Name	1-4	A4
b.	Input Node	5-8	A4
c.	Output Node	9-12	A4
d.	Distribution	13	I1

In column 13, only integers 1, 2, 3, and 4 may be entered.

These integers are used to indicate one of four possible distribution types:

Integer	Distribution Type
1	Normal
2	Triangular
3	Uniform
4	Constant

e.	First time distribution argument	14-23	F10.0
f.	Second time distribution argument	24-33	F10.0
g.	Third time distribution argument	34-43	F10.0

When the distribution is normal, the first time distribution argument is the mean and the second time distribution argument is the standard deviation. The third time distribution argument is left blank.

When the distribution is triangular, the first time distribution is the most optimistic value, the second time distribution argument is the most likely value and the third time distribution argument is the most pessimistic value.

When the distribution is uniform, the first time distribution argument is the optimistic value, the second is the pessimistic value and the third is blank.

In a constant distribution, the first time distribution argument is the constant value and the second and third arguments are blank.

h.	Fixed cost	44-53	F10.0
i.	Variable Cost Coefficient	54-63	F10.0
j.	Probability of successful completion	64-73	F10.0

Both the fixed cost and variable cost coefficients are components of a linear equation which describes the cost as a function of time. This equation is written:

Cost = fixed cost + variable cost coefficient x t
where t is time in appropriate units.

The probability of successful completion refers to the probability that an activity (arc) will reach its desired end (node).

The arc cards which comprise Block II may be inputed in any order. However, it is desirable to enter the arcs in the order in which they appear in the network to avoid confusion.

The end of Block II is indicated by a card following the last arc card in Block II with "RETU" in card columns 1-4.

Block III: The information required for each node is formatted as follows:

	Info	Columns	Format
a.	Node name	1-4	A4
b.	Input rule	5	I1

There are 6 possible integers (input rules) which appear in column 5:

Integer Value	Input Rule
1	And
2	Or
4	Initial
5	1/1
6	$\overline{1/1}$
7	Preferred

c. Output rule	6	I1

There are 6 possible integer values which can appear in column 6. Listed below is each integer and the indicated output rule:

Integer Value	Output Rule
1	All
2	Prob
4	Terminal
5	1/1
6	1/1 Bar
7	Preferred

When the node card has an input/output rule 1/1, $\overline{1/1}$ or PREF, a second card with the following information is required:

Info		
a. No. of Output Arcs	1-2	I2
b. Input Arc Names	3-6, 11-14, 19-22, etc.	A4
c. Output Arc Names	7-10, 15-18, 23-26, etc.	A4

Each input arc name and its corresponding output arc name will appear as pairs on this card. The first input arc name will appear first in columns 3-6 followed by the output arc name it initiates (columns 7-10). The second input and output arc pairs will appear in columns 11-14 and 15-18 respectively, and so on until all arc pairs are listed.

Since the default output arc has no corresponding input arc name ($\overline{1/1}$ and PREF), "ZZZZ" must be entered as the input arc name. The above information card must immediately follow the node card it describes.

When the node card has an output rule Prob (2), a second card with the following information is required:

Info	Columns	Format
a. No. of Output Arcs	1-2	I2
b. Output Arc Names	3-6, 13-16, 23-26, etc.	A4
c. Probabilities associated with	7-12, 17-22, 27-32, etc.	F6.3

On this card, the name of each output arc and the probability of its exiting appear in pairs similar to the input and output arc pairs of the previous card. For example, the name of the first output arc will appear in columns 3-6 followed by the probability that it will exit in columns 7-12 and so on until all arc name and probability pairs have been listed. This card must immediately follow the node card it describes.

The node cards in Block III may be <u>inputed</u> in any order. However, be certain that in the case of 1/1, $\overline{1/1}$, Preferred and Probability nodes that each node card is succeeded by its appropriate information card. Following the last node card in Block III is a card with "RETU" in card column 1-4 marking the end of the block.

In the next section a hypothetical problem will be described, structured and inputs prepared as a network for RISCA.

3.4 <u>Example Problem.</u>

The use of the program will be illustrated via an example problem containing a variety of arc and node types.

A worker is confronted with a new tardiness policy established by his employer. It has been decided that an employee who is late for work more than 10 percent of the next 500 working days will be docked accordingly. The amount that the worker will be docked will be the cumulative time lost due to lateness.

This announcement has prompted the worker to closely examine the various routes and hazards facing him each morning in order to evaluate his chances of his being docked.

His analysis of the paths* for driving to work revealed the following critical areas:

a. A fork in the road about 12 minutes from home. It has been his experience that the shortest route is only possible 90 percent of the time due to hazardous road conditions. 10 percent of the time he must travel a considerably longer route.

b. Fuel Problems. The worker is assured of not running out of gas if he takes the shorter route. However, if he takes the longer, alternate route, there is an 80 percent chance that he will have to make a 5 minute fuel stop.

c. Rough Road. Because of the extreme punishment to his tires along a stretch of road beyond the service station, the worker feels that he has a 5 percent chance of getting a flat tire that would take 15 minutes to repair.

d. Rider Stop. Beyond the stretch of bumpy road is the home of the worker's friend who rides to work with him every day. Historical data reveals that:

(1) If the shortest route is taken, and no fuel stops or tire troubles are experienced, his rider will be waiting outside for a ride 100 percent of the time.

(2) If the alternate route is taken, and no fuel stops or tire troubles are experienced, his rider will be waiting outside 80 percent of the time.

(3) If he takes the alternate route, stops for gas and has no tire trouble, his rider will be waiting for him 60 percent of the time.

---

* Recall that a path is a sequence of activities connecting two events. In this instance the events are departing for work and arriving at work.

(4) If he has a flat tire, regardless of previous routes or stops, his rider will only be waiting outside 40 percent of the time.

In each case when the rider is not outside waiting, he must stop and check to see whether he is inside or has obtained another rider. This is a 5 minute stop.

Figure 3.1 illustrates the alternative routes and hazards.

In order to determine which node and arc type should be used to simulate the route in a network, each critical area should be examined.

a. Fork in the road. This is clearly a probabilistic node. The probability that the arc representing the shorter route will emanate is .90 and the probability that the arc representing the longer route will emanate is .10.

b. Gas Station. A $\overline{\text{I/I}}$ node will be used to simulate this event although a probabilistic node could also be used. The arc entering this node represents the longer route path and a probability of completion of .8 will be assigned to it. In the event that the arc is not completed, the default arc representing a gas stop will be initiated.

c. Rough Road. Three $\overline{\text{I/I}}$ nodes will be used to represent this hazard. Three nodes are used instead of one in order to account for this event in each of the three deterministic subnetworks developed thus far. Which $\overline{\text{I/I}}$ node is used in each iteration is dependent upon which of the following alternatives precedes it:

(1) Short route,

(2) Long route, no gas stop,

(3) Long route, gas stop.

An arc, representing each of these alternatives, will be assigned a probability of completion of .95. In each case, failure to complete the node (get through the bumpy road without a flat) will initiate a default arc. All three default arcs will enter the fix flat node which will have an OR input rule.

d. Rider Stop. This event will be represented by three I/I nodes and an AND/ALL node. The AND/ALL node represents picking up the rider when the short route is taken and there is no tire trouble. Recall that in this case, the rider is waiting outside 100 percent of the time.

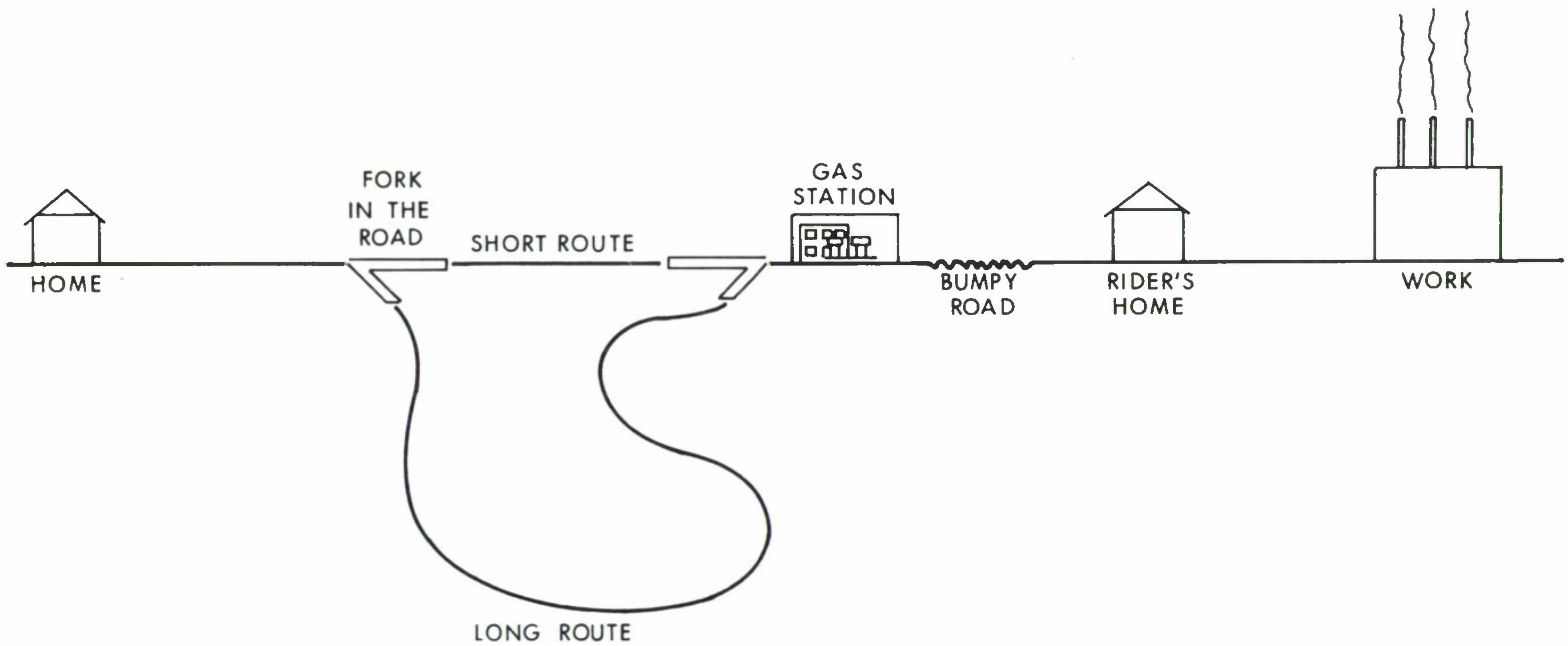


Figure 3.1 Graphical Representation of the Driving to Work Problem.

In the other three instances, the probability that the rider will be waiting is less than 1.0. These probabilities are

	Event	Probability
(1)	Long route, no gas stop, no tire trouble	.8
(2)	Long route, gas stop, no tire trouble	.6
(3)	Tire trouble	.4

Separate $\overline{I/I}$ nodes will be used in each of these instances with all three default arcs entering an OR/ALL node representing a stop to check on the rider.

e. Arrive at Work. Five AND/TERM nodes will be used to represent the worker's arrival. Their time and probability values will represent the time and probability associated with 5 separate alternatives which are:

(1) Short route, no stops,

(2) Long route, no stops,

(3) Long route, gas stop,

(4) Stop to fix flat,

(5) Stop for rider.

Figure 3.2 illustrates the network representation of "Driving to Work" using the RISCA and MATHNET arc and node construction symbols.

Table 3.2 lists each of the time distribution arguments, the events to which they correspond, the distribution arguments, the events to which they correspond, the distribution type involved and the arc names assigned to each of these time consuming activities.

Now that the network has been graphically represented, the arcs and nodes have been labeled and the time distribution argument has been determined, the input deck can be prepared.

As explained earlier, there are some differences in MATHNET and RISCA with respect to input deck content.

The contents of the RISCA deck are the three blocks outlined previously. That is, the first card is BLOCK I consisting of one title card describing the network to be analyzed. This is followed by

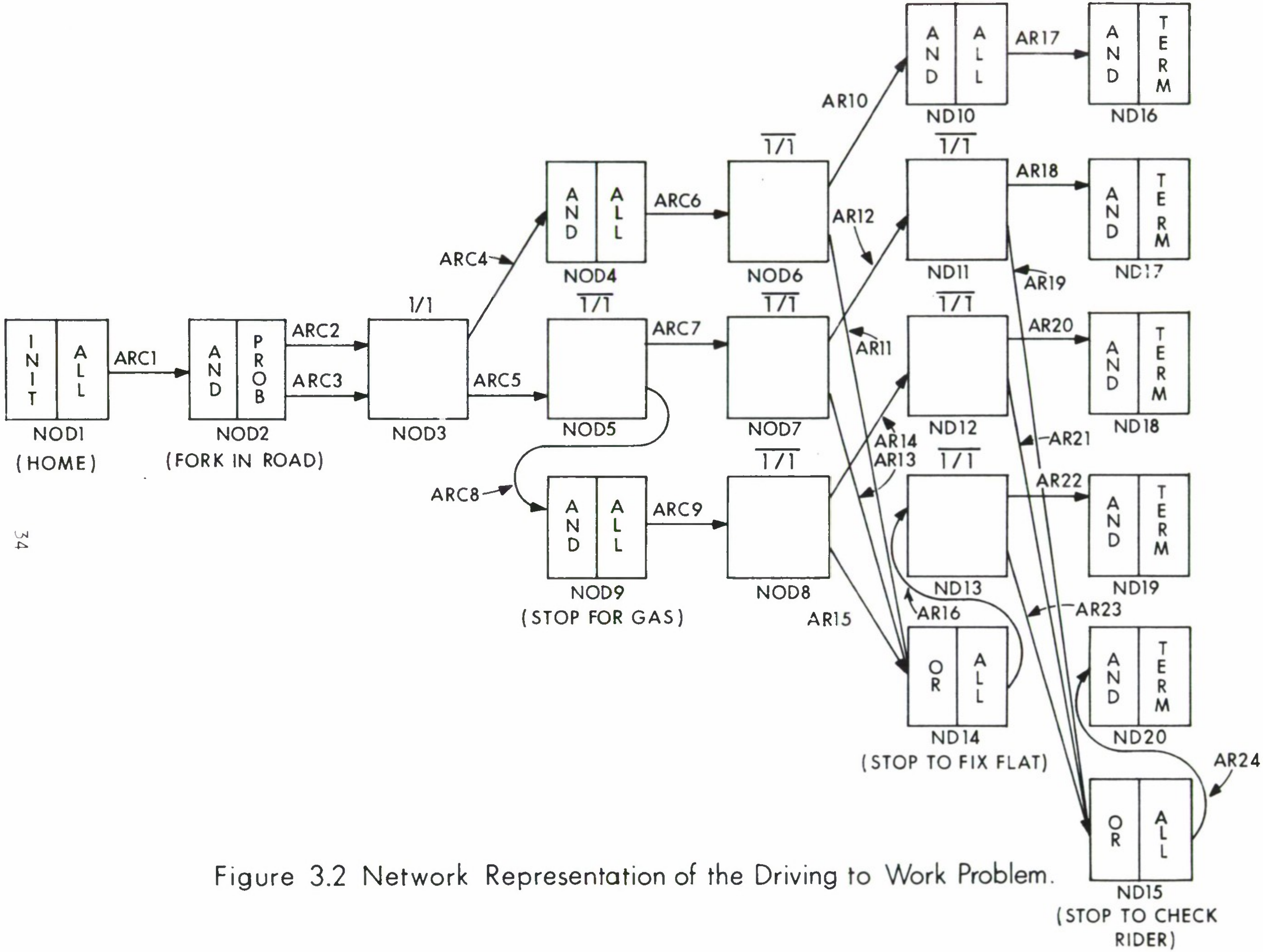


Figure 3.2 Network Representation of the Driving to Work Problem.

TABLE 3.2 CHARACTERISTICS OF ARCS USED IN EXAMPLE PROBLEM

Activity	Distribution Type	Arc Name	Time Distribution Arguments		
			Optimistic	Expected	Pessimistic
From Home to Junction	Triangular	Arc1	10.0	12.0	14.0
Junction to Gas Station (short route)	Triangular	Arc2	5.0	6.0	7.0
Junction to Gas Station (long route)	Triangular	Arc3	11.0	13.0	15.0
Gas Station Stop	Constant	Arc8	5.0		
Gas Station to Bumpy Road	Triangular	Arc6	3.0	4.0	5.0
		Arc7	3.0	4.0	5.0
		Arc9	3.0	4.0	5.0
Stop to Change Flat Tire	Constant	AR15	15.0		
Bumpy Road To Rider's House	Triangular	AR10	4.0	5.0	6.0
		AR12	4.0	5.0	6.0
		AR14	4.0	5.0	6.0
		AR16	4.0	5.0	6.0
Stop to Check Rider	Constant	AR23	5.0		
Rider's House To Work	Triangular	AR17	8.0	9.0	10.0
		AR18	8.0	9.0	10.0
		AR20	8.0	9.0	10.0
		AR22	8.0	9.0	10.0
		AR24	8.0	9.0	10.0

BLOCK II, the arc cards, the last card being a "RETU" and indicating termination of the block. Then comes BLOCK III, the node cards, the last card again being a "RETU" card indicating the end of the block.

The version of RISCA described in this manual is designed only to run in a batch mode while MATHNET is designed to run in either a batch or time shared mode. It should be emphasized that adapting RISCA for a time sharing environment is not a difficult task. However, it would probably not be possible to run these versions of MATHNET and RISCA in any batch processing mode without first adapting the program to the particular system. Similarly, it would probably not be possible to run this version of MATHNET in any time sharing system without adapting it to the system. Because the MATHNET program can be run in either mode, the input deck is designed to give operators working in time-share the capability of introducing either each section of the input deck with a card describing the section or a card describing an operation to be performed. When the program reads the introduction card, it is directed to that portion of the program which performs the indicated operation.

The first card in the MATHNET input deck is a card with either a 0 or a 1 punched in column 1 (format I1). The 0 card indicates that the problem will be run in a time share environment, and a 1 indicates batch mode.

For the purposes of comparing RISCA and MATHNET inputs in this manual, only the input decks for a batch mode environment will be utilized (See Figure 3.3).

The information cards mentioned above are cards with an integer punched in column 1 (format I1). The following is a list of each integer which may be used, and the information card it indicates:

Integer	Indicates
1	Node cards follow
2	Arc cards follow
3	Net iteration number is set
4	Net is scanned
5	Net is run
6	Identification card follows
9	Run is to be completed (Ends Session)

RISCA INPUT DECK

```
THE NEW DRIVING TO WORK PROBLEM
ARC1ND01ND022   10.00   12.00   14.00   1.00
ARC2ND02ND032    5.00    6.00    7.00   1.00
ARC3ND02ND032   11.00   13.00   15.00   1.00
ARC4ND03ND043                           1.00
ARC5ND03ND053                            .80
ARC6ND04ND062    3.00    4.00    5.00    .95
ARC7ND05ND072    3.00    4.00    5.00    .95
ARC8ND05ND093    5.00    5.00    5.00   1.00
ARC9ND09ND082    3.00    4.00    5.00    .95
AR10ND06ND102    4.00    5.00    6.00   1.00
AR11ND06ND143   15.00   15.00   15.00   1.00
AR12ND07ND112    4.00    5.00    6.00    .80
AR13ND07ND143   15.00   15.00   15.00   1.00
AR14ND08ND122    4.00    5.00    6.00    .60
AR15ND08ND143   15.00   15.00   15.00   1.00
AR16ND14ND132    4.00    5.00    6.00    .40
AR17ND10ND162    8.00    9.00   10.00   1.00
AR18ND11ND172    8.00    9.00   10.00   1.00
AR19ND11ND153    5.00    5.00    5.00   1.00
AR20ND12ND182    8.00    9.00   10.00   1.00
AR21ND12ND153    5.00    5.00    5.00   1.00
AR22ND13ND192    8.00    9.00   10.00   1.00
AR23ND13ND153    5.00    5.00    5.00   1.00
AR24ND15ND202    8.00    9.00   10.00   1.00
RETU

ND0141
ND0212
 2ARC2   .90ARC3   .10
ND0355
 2ARC2ARC4ARC3ARC5
ND0411
ND0566
 2ARC5ARC7ZZZZARC8
ND0666
 2ARC6AR10ZZZZAR11
ND0766
 2ARC7AR12ZZZZAR13
ND0866
 2ARC9AR14ZZZZAR15
ND0911
ND1011
ND1166
 2AR12AR18ZZZZAR19
ND1266
 2AR14AR20ZZZZAR21
ND1366
 2AR16AR22ZZZZAR23
ND1421
ND1521
ND1614
ND1714
ND1914
ND2014
RETU
```

CONTROL CARDS (MATHNET ONLY)

TITLE CARD (RISCA AND MATHNET)

CONTROL CARDS (MATHNET ONLY)

ARC CARDS (RISCA AND MATHNET)

CONTROL CARD (MATHNET ONLY)

NODE CARDS (RISCA AND MATHNET)

CONTROL CARDS (MATHNET ONLY)

MATHNET INPUT DECK

```
1
5
THE NEW DRIVING TO WORK PROBLEM
3
0500
2
ARC1ND01ND022   10.00   12.00   14.00   1.00
ARC2ND02ND032    5.00    6.00    7.00   1.00
ARC3ND02ND032   11.00   13.00   15.00   1.00
ARC4ND03ND043                           1.00
ARC5ND03ND053                            .80
ARC6ND04ND062    3.00    4.00    5.00    .95
ARC7ND05ND072    3.00    4.00    5.00    .95
ARC8ND05ND093    5.00    5.00    5.00   1.00
ARC9ND09ND082    3.00    4.00    5.00    .95
AR10ND06ND102    4.00    5.00    6.00   1.00
AR11ND06ND143   15.00   15.00   15.00   1.00
AR12ND07ND112    4.00    5.00    6.00    .80
AR13ND07ND143   15.00   15.00   15.00   1.00
AR14ND08ND122    4.00    5.00    6.00    .60
AR15ND08ND143   15.00   15.00   15.00   1.00
AR16ND14ND132    4.00    5.00    6.00    .40
AR17ND10ND162    8.00    9.00   10.00   1.00
AR18ND11ND172    8.00    9.00   10.00   1.00
AR19ND11ND153    5.00    5.00    5.00   1.00
AR20ND12ND182    8.00    9.00   10.00   1.00
AR21ND12ND153    5.00    5.00    5.00   1.00
AR22ND13ND192    8.00    9.00   10.00   1.00
AR23ND13ND153    5.00    5.00    5.00   1.00
AR24ND15ND202    8.00    9.00   10.00   1.00
RETU
1
ND0141
ND0212
 2ARC2   .90ARC3   .10
ND0355
 2ARC2ARC4ARC3ARC5
ND0411
ND0566
 2ARC5ARC7ZZZZARC8
ND0666
 2ARC6AR10ZZZZAR11
ND0766
 2ARC7AR12ZZZZAR13
ND0866
 2ARC9AR14ZZZZAR15
ND0911
ND1011
ND1166
 2AR12AR18ZZZZAR19
ND1266
 2AR14AR20ZZZZAR21
ND1366
 2AR16AR22ZZZZAR23
ND1421
ND1521
ND1614
ND1714
ND1814
ND1914
ND2014
RETU
4
5
9
7
```

Figure 3.3 Comparison of RISCA and MATHNET Input Decks for the Example Problem

Note that the number of iterations is controlled by the operator in MATHNET. In RISCA, the number of iterations is fixed at 500. It is conceivable that only 500 iterations could be restrictive in a network having a large number of terminal nodes. However, as was mentioned above, the program can be easily expanded to allow for more iterations. The card following the information card with a 3 in column I is a card with an integer value in columns I-5 (format I5) specifying the number of iterations (from I to I000).

The information cards with integers 4, 5 and 9 punched in column I do not introduce input cards, but they indicate to the program that one of three operations are to be performed. These operations are:

a. Scan the net - A scan of the net means that all arc and node characteristics are printed out in a tabular format (see Section 4).

b. Run the net - The net simulation is done.

c. Run is Completed - All output from the simulation is printed out. (See Section 4).

The arc, node and identification cards are formated in the same manner as RISCA. However, the order in which they are inputted is not as restrictive as RISCA. The data cards must be entered before the operation cards, but the order in which the data cards are entered is not important. The operation cards, however, must be entered as follows:

a. Scan the net

b. Run the net

c. Complete the run.

Figure 3.3 provides a comparison of the RISCA and MATHNET input decks used in the example problem. This figure illustrates the fact that with the exception of the MATHNET control cards discussed above, the two decks are identical.

Since the RISCA input deck has already been discussed in detail, and the fact that its MATHNET counterparts are identical has been established, only an explanation of the MATHNET control cards in the example problem input deck remains to be covered.

The first control card indicates that the problem will be run in a batch mode environment (I in column I). The second card indicates that the problem identification card follows (6 in column I). The third control card (3 in column I) indicates that the number of iterations follows. The next card sets the number of iterations at 500. The following card indicates that the arc cards follow (2 in

column 1). The next control card appears at the end of the arc card block and indicates that the node cards follow (1 in column 1).

The last 3 cards in the MATHNET input deck control the running of the simulation. The first card initiates the scanning of the net (4 in column 1). The second card indicates the simulation is to be run (5 in column 1). The last card indicates that the run is to be completed and the results printed out (9 in column 1).

It is important to note that the blanks which appear in the listing of the RISCA input deck in figure 3.3 were included for ease of comparison only and are not required.

It was pointed out above that there are certain restrictions as to the number of iterations permissable in these versions of MATHNET and RISCA. In addition to these iteration restrictions, there are certain other inherent restrictions which deserve mention. For a complete listing, see Table 3.3. These limits may be increased (through minor program modifications) to the upper bound of the computer memory core.

4. RISCA AND MATHNET PROGRAM OUTPUT

4.1 Introduction.

In this section, the output from MATHNET and RISCA for the going-to-work example will be described and compared. In addition, the potential uses of the output for decision making purposes will be discussed both in and out of the context of this example.

4.2 MATHNET Output.

The output of any MATHNET run consists of a detailed input listing, frequency histograms for completion time and cost for each of the terminal nodes, frequency histograms of completion times and cost weighted over all terminal nodes, and frequency histograms of the percent of time each terminal node was selected in the simulation. A complete output listing for MATHNET for the example is provided in Appendix IV.

The first four pages of output consists of a printout of the dialogue between the user and the program, and a detailed description of the network representation being simulated. (See the Input Section). This dialogue is really only applicable to the terminal user. It provides him with a detailed set of instructions for inputting his network representation. This example was run in a batch mode, therefore the first data card inputted into the program had to have a 1 in the first column of the first card. On the two pages following the dialogue there is a detailed listing of the arc and node input data. This is particularly

TABLE 3.3 Restrictions to MATHNET and RISCA

	MATHNET	RISCA
Maximum number of nodes	100	100
Maximum number of arcs	500	100
Maximum number of arcs into, or out of a single node	10	10
Maximum number of initial nodes	10	10
Maximum number of terminal nodes	30	30
Maximum number of iterations	1000	500

useful information for checking the network. The format of the arc and node data will not be discussed since this is the same format that was discussed in the Input section.

Next, for each of the terminal nodes (ND 16, 17, ND 18, ND 19, ND 20) a frequency histogram* of the completion time is provided. In this example, costs were not considered, but if they were then there would be a frequency histogram of cost for each terminal node. In addition, each terminal node represents the event "arriving at work" however, in each instance the path (i.e. sequence of events) is different. For example, the first histogram is for the completion times for ND 16. ND 16 is the short route to work with no time delay associated with hazards, stopping for gas, or picking up a rider. The horizontal axis is the frequency of occurrence and the vertical axis is time in minutes.** Looking at this graph, the probability of arriving at work within 35.6 and 35.9 minutes given this route is taken is .087. The interpretation of all the remaining frequency histograms is the same with two exceptions. One is that if the terminal cost frequency histograms were given then vertical axis would be cost. The other is that the final pair of frequency histograms for completion times and cost (where considered) is weighted over all possible terminal nodes (routes).

The final graph provided is the frequency plot of the percentage of times each terminal node was selected in the simulation. The horizontal axis again represents frequency of occurrence and the vertical axis lists the possible terminal nodes. Looking at the graph in Appendix IV, the probability of arriving at work having taken route ND 17 is .046, or ND 17 was chosen in 4.6 percent of the simulations.

4.3 RISCA Output.

The output of any RISCA run consists of a detailed input listing, frequency and cumulative frequency histograms for completion times and cost for each of the terminal nodes, frequency and cumulative frequency histograms of completion times and cost weighted over all terminal nodes, and a frequency histogram of the percent of times each terminal node was selected in the simulation. A complete listing of RISCA for this example is provided in Appendix V.

---

*On each time and cost frequency histogram the following statistics are provided: mean, variance, median, and mode.

**The time and cost units are selected by the analyst and depend upon the problem.

The first page of output consists of a detailed listing of the arc and node input data. This is particularly useful information for checking the network. Once again, the format of the arc and node data will not be discussed since this is the same format that was discussed in the Input Section.

Next, for each of the terminal nodes (ND 16, ND 17, ND 18, ND 19, ND 20) a frequency and cumulative frequency histogram* of the completion times and cost is provided. It is emphasized again that costs were not considered. In addition, each terminal node represents the event arriving at work, but in each instance the path (i.e., sequence of events) is different. Since the interpretation of the frequency histograms is the same as for MATHNET, only the cumulative frequency histograms will be discussed. For example, the first histogram is for the completion times for ND 16. It is followed by the cumulative frequency histograms for completion times for ND 16. The vertical axis is time** to completion, and the horizontal axis is the probability that the true time is less than or equal to the time on the vertical axis. For instance, the probability of arriving at work in less than 35.647 minutes given that this path was taken is .401. The interpretation of all of the remaining frequency and cumulative frequency histograms is the same with two exceptions. One is that for cost frequency histograms, the vertical axis is cost**. The other is that the final pair of frequency and cumulative frequency histograms for completion times and cost is weighted over all possible terminal nodes (routes).

The final graph provided is the frequency plot of the percent of time each terminal node was selected in the simulation. The interpretation of this histogram is identical to the MATHNET interpretation of the same histogram.

4.4 MATHNET and RISCA Output Comparison.

There are only two differences in the output of these programs. One is that RISCA provides cumulative frequency histograms for time and cost, for terminal nodes, and MATHNET does not. Clearly, the cumulative frequency histogram is required information in most applications as for example, in estimating the probability of meeting program time and cost goals. Consequently, there is an advantage in using this version of RISCA over this version of MATHNET since it eliminates the need for manually generating cumulative frequency histograms.

---

*On each time and cost histogram and following statistics are provided: mean, variance, and standard deviation.

**The time and cost units are selected by the analyst and depend upon the problem.

The other difference is in the output statistics computed for each time and cost distribution. In MATHNET, the mean, variance, median and mode are computed while in RISCA the mean, variance and standard deviation are computed. On first glance it may appear that one is given more information with the MATHNET statistics, but this is not the case. Since, if one is given the frequency and cumulative frequency histogram for any terminal node, determining the mode and median is a minor operation.

4.5 Use of the Output.

While there are more ways to use the output of a network analysis than will be discussed in this section, the two decision problems that are discussed probably represent the more popular applications. Before describing these problems and how to use the output for decision making purposes several general comments should be reiterated.

First, the greatest benefit to be derived in using a network analyzer program comes from the effort that is put into modeling the project or system as a network. Using this type of tool forces one to examine all possible events and the interaction of these events in the program. Further, representing a system as a network allows one to handle all of the relevant decision information in a systematic and composite fashion and to evaluate the impact of interactions that would be otherwise impossible.

Next, no matter what the decision is, there will always be a need for an overall framework for consolidating all the information. Network analyzer programs such as MATHNET and RISCA provide such a framework.

Finally, once the network has been structured, future modifications to the network are a relatively simple matter. In many instances, this type of analysis should not be a one time effort. The network representation and time and cost estimates should be modified on a periodic basis because as time passes, more information is gained and the initial network may no longer realistically represent the system.

In the remainder of this section, the use of network analysis for two general decision problems and the going to work example will be discussed.

The first decision problem either is one where the program* status must be evaluated periodically to determine whether or not to

---

* Program refers to a development program for a system.

continue or one where the impact of a program change must be evaluated. The discussion that follows will only be in terms of the decision to continue for brevity purposes, although network analysis could be used to help in a program change decision. A good example of a program would be a weapon system development. Clearly, the decision to continue depends on the chances of successfully developing the system within time and budget constraints.

A network representation for a development program would probably consist of two sets of outcomes. One set would constitute successful development, and the other set would constitute failure. In this case, the chances of successful development equals the sum of the percent of times each successful terminal was selected in the simulation. For any successful outcome the probability of developing the system within time and cost constraints may then be evaluated. This information is taken directly from the RISCA output (i.e. frequency histogram of the percent of times the terminal nodes were selected and the appropriate terminal node cumulative time and cost histograms). In addition, without a great deal of effort, the frequency and cumulative frequency histograms for time and cost weighted over all successful outcomes are estimated. This is done by either modifying the network representation so that all successful terminal events feed into one terminal event (successful development), or it may be constructed from the detailed frequency histograms of the successful outcomes.

This type of information should give the decision maker a foundation upon which to decide whether to continue with the program in light of the risks. Of course this doesn't relieve the burden of decision making, but it should enable the decision maker to make a more informed decision. Further, as the program continues, the network could be updated periodically to evaluate the program status for any future program decisions.

The other decision problem is one in which a choice between alternative systems for meeting a particular set of requirements* must be made. For this type of problem, a network could be structured for each alternative.** Each of these networks could then be simulated and the chances of successfully developing each alternative system and the corresponding development costs and completion times could be estimated. Given this information, the analyst has several relative measures for comparing the alternatives, and the decision maker should

---

*This assumes that "meeting a particular set of requirements..." generates equal and required effectiveness.

**It is assumed that a system is to be developed.

be able to systematically consider time and cost trade-offs in light of the risks.

Even though these two hypothetical decision problems are over-simplified, they are representative of many of the decisions that must be made within AMC. In both of these decision problems there is a great need for an overall framework for the evaluation. All too frequently this framework seems to be lacking. Network analysis provides an overall framework for synthesizing a large portion of the decision information in a systematic fashion that takes into account the known uncertainties. It is not meant to imply that the network analysis is the answer to all of the decision makers problems, but reasonable applications are a step in the right direction.

Returning now to the going to work example: How can this analysis help the worker in achieving his objective? First, it is assumed that the objective is not to be docked for tardiness. Since in all instances the worker arrives at work, the distribution of interest is the arrival time distribution weighted over all possible outcomes. Assuming the workman continues to leave his house at 0705 every morning and work starts at 0745, the probability of his being late is .157 or 1.0 minus the probability of arriving at work in less than or equal to 40 minutes (.843) (See the cumulative frequency histogram for arrival times weighted over all outcomes in Appendix V.)

If the probability of being late on any given day that the worker departs at 0705 is .157, the risk of being late over 10 percent of the next 500 working days is the probability of being late 51 or more times. Let ℓ be the random variable representing the number of late arrivals and p the probability of being late on a given day, then

$$P[51 \leq \ell \leq 500] = \sum_{\ell=51}^{500} \binom{500}{\ell} p^{\ell}(1-p)^{500-\ell}$$

is the risk of being late in excess of 10 percent of the time.

If the worker leaves at 0705, the above calculation results in a risk of .99(p=.157).

Using this method it is possible to derive a risk-of-being-docked profile for the worker as a function of departure time (See Figure 4.1).

What should the worker do? The answer is not clear cut. It depends upon the worker's assessment of the value of additional rest versus the potential financial impact of being docked. However, given the risk profile, the worker is in a position to consider the trade-offs. For example, if the worker cannot possibly afford to be docked, he can select a departure time from the risk profile where the risk is approximately zero (i.e., leave at 0654 or earlier).

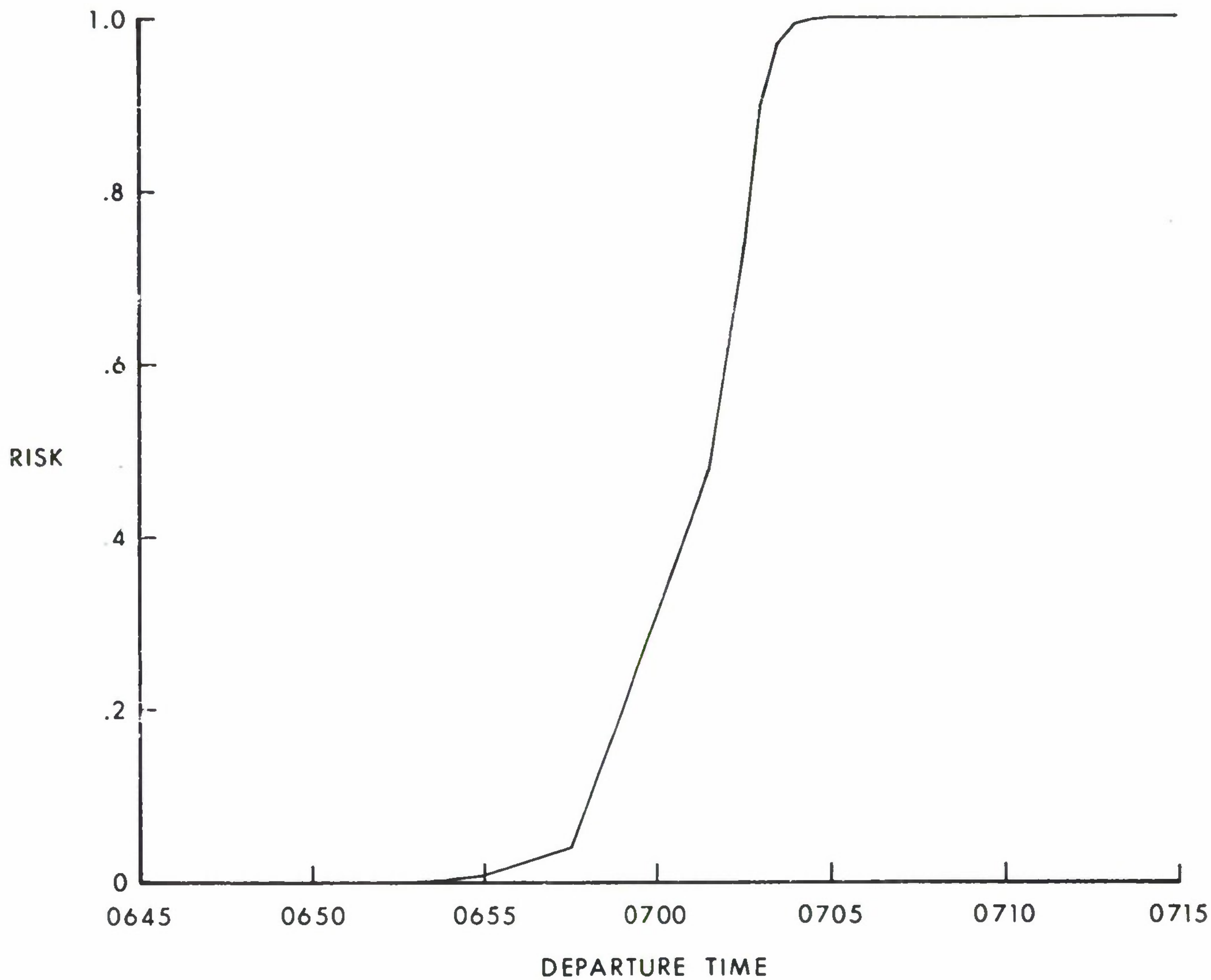


Figure 4.1 Risk Profile

Thus, it is seen in this simple example that, through the modeling of a network using RISCA or MATHNET, the analyst gains significant insight into the risks inherent in a system composed of various alternatives. This insight into the risks provides the information for making decisions concerning which alternative to choose, or decisions involving the various cost and/or time parameters (in this example departure time is the deciding factor).

Of course, much more complex systems containing a myriad of activities, events and alternatives may be modeled using these modeling tools. The scope and complexity of each network is only constrained by the number of arcs and nodes that can be inputted.

5. SUMMARY

MATHNET and RISCA are programs for simulating a class of networks where both the events and activities can be modeled probabilistically. These programs should provide the analyst with a tool for evaluating the status of existing programs, for evaluating the impact of proposed program changes on the total program, for deciding whether to continue the program, and for deciding between alternative systems. These decisions may not be mutually exclusive.

Since RISCA is a modification of MATHNET, there are very few differences in the input and output of the programs, and there are no differences in the method of simulating the networks. As seen in the Output Section, the additional features of this version of RISCA's output to this version of MATHNET's output are not really significant.

Throughout this report R&D applications have been emphasized, however, these techniques obviously have broader applications.

Next page is blank.

BIBLIOGRAPHY

A Course of Instruction in Risk Analysis, ALM-3315-H, United States Army Logistics Management Center, Fort Lee, Virginia, UNCLASSIFIED.

MATHNET Mathematica's Network Analyzer Program, Preliminary Edition, August 1970, Mathematica, Princeton, New Jersey, UNCLASSIFIED.

Next page is blank.

APPENDIX I

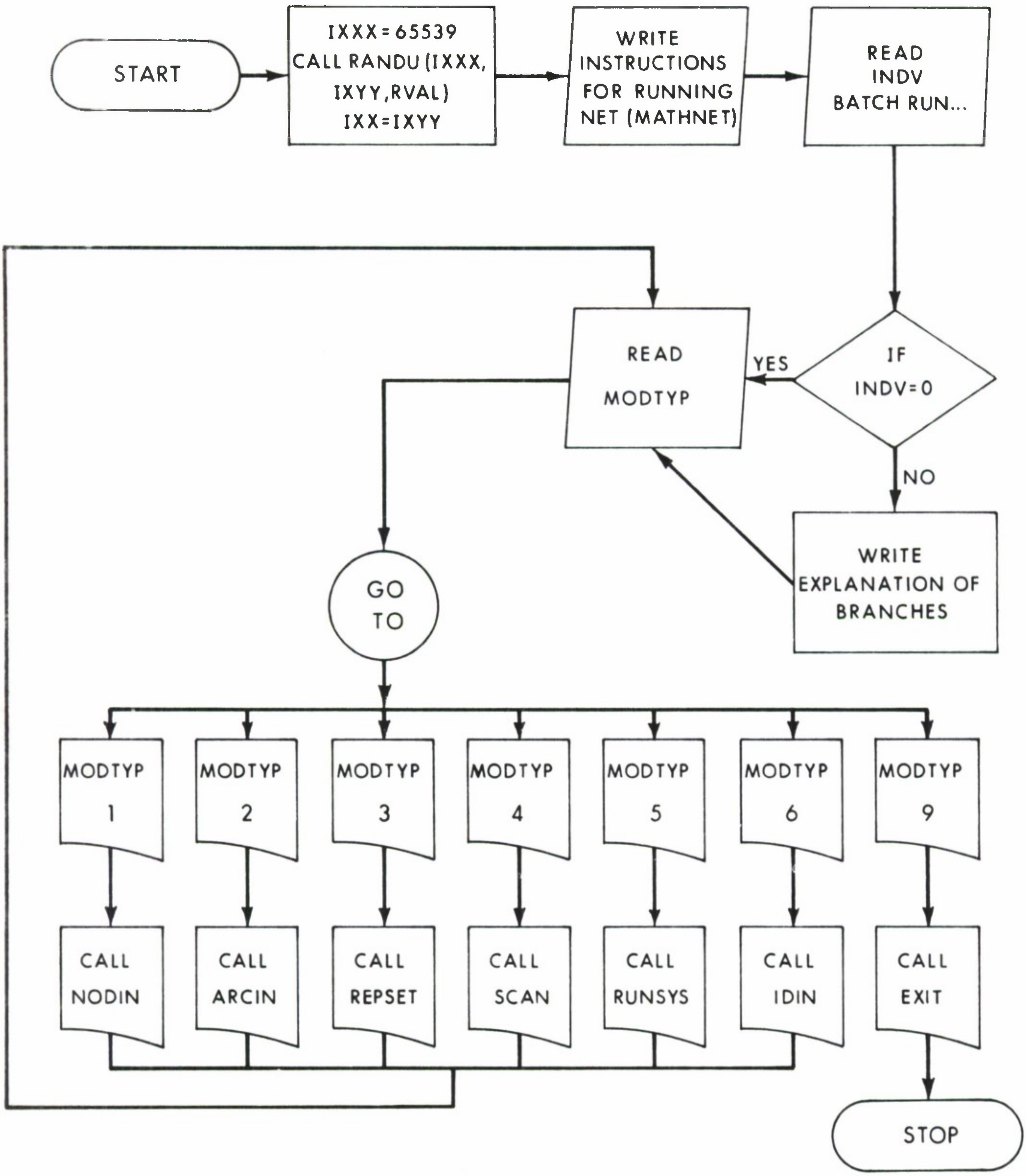
GENERAL FLOW CHART AND LISTING OF MATHNET

Next page is blank.

This appendix includes a general processing flow chart for MATHNET, but it does not include a description or flow chart of the individual subroutines. A detailed description and flow chart of the individual subroutines are provided in Appendix III. In addition to the flow chart, a complete program listing is provided. This version of MATHNET is designed to be run in either a batch or time sharing mode.

MATHNET is written in FORTRAN IV. Even though FORTRAN IV is considered to be standard language, adapting the program for a particular computer will probably require minor program modifications. These modifications generally result from pecularities of the given system.

There is one definite modification that the user must make before MATHNET can be run on his computer. To run the simulation it is necessary to generate uniform random numbers. Within the subroutine RANDU these random numbers are generated by calling the library subroutine peculiar to the particular machine. In this listing the library subroutine RANSET is called to generate the uniform random numbers. Therefore, the user must either call the appropriate uniform random number generator for his machine or the analyst must write his own uniform random number generating routine within the RANDU subroutine. The latter was done by ALMC in their modification of MATHNET. The interested reader is referred to Appendix II where the ALMC Program is listed.



Flow Chart of "MATHNET"

```
      PROGRAM MATHW ( INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,
     +  PUNCH,TAPE7=PUNCH,TAPE1,TAPE2,TAPE3,TAPE4 )
C     *******************************************************************
C     *******************************************************************
C
C     MATHNET WAS DEVELOPED BY MATHEMATICA INC. FOR THE ARMY
C     RESEARCH OFFICE UNDER CONTRACT NUMBER DAHC 04 70 C0025.
C     ALTHOUGH THIS PROGRAM HAS BEEN EXTENSIVELY TESTED WITH A LARGE
C     NUMBER OF NETWORKS,EXHAUSTIVE TESTING OF ALL POSSIBLE CLASSES
C     OF NETWORKS IS COMPUTATIONALLY IMPRACTICAL.MATHEMATICA THEREFORE
C    MAKES NO GUARANTEES CONCERNING THE ACCURACY OF THE OUTPUT GENERATED
C     BY A RUN OF THE MATHNET PROGRAM
C     IT WOULD BE APPRECIATED IF ANY NETWORKS WHICH PRODUCE ERRONEOUS
C     OUTPUT,OR ANY OTHER SYSTEM ERRORS, WOULD BE REPORTED TOO
C     MR. STEPHEN ROBINSON
C     MATHEMATICA
C     ONE PALMER SQUARE
C     PRINCETON,NEW JERSEY  08540
C
C     SUGGESTIONS CONCERNING ADDITIONAL OUTPUT FEATURES,ARC
C     CHARACTERISTICS,OR NODE CAPABILITIES,ARE ALSO WELCOMED.
C
C     *******************************************************************
C     *******************************************************************
C
C     MATHNET - VERSION 1 MODIFICATION LEVEL 0 - AUGUST 1, 1970
C
C     *******************************************************************
C     *******************************************************************
C
C
C     THIS IS THE MAIN ROUTINE
      COMMON/RANC/IXX
      COMMON/INDEV/ INDV
C
C     INITIALIZE RANDU
      IXXX=65539
      CALL RANDU(IXXX,IXYY,RVAL)
      IXX=IXYY
      WRITE(6,20)
C
C     FIRST I WILL READ IN A RECORD FROM DATA SET 5 TO DETERMINE
C     IF WE ARE IN BATCH MODE OR NOT
C
 20   FORMAT(1X, 39HIF YOU ARE RUNNING THIS FROM A TERMINAL)
      WRITE(6,21)
 21   FORMAT(1X, 45HPLEASE ENTER A 1, IF RUNNING BATCH YOU SHOULD)
      WRITE(6,22)
 22   FORMAT(1X, 28HHAVE ENTERED A CARD WITH A 0)
      WRITE(6,23)
 23   FORMAT(1X, 12HFORMAT IS I1)
      READ(5,24) INDV
 24   FORMAT(I1)
C     NOW TEST MODE, IF BATCH SKIP EXPLANATIONS FROM NOW ON
      IF(INDV.EQ.0) GO TO 7
      WRITE(6,30)
 30   FORMAT(1X, 53HYOU ARE NOW IN MONITOR MODE , FROM THE FOLLOWING LIS
     CT)
      WRITE(6,31)
 31   FORMAT(1X, 35HSELECT THE MODE YOU WISH TO GO INTO)
      WRITE(6,32)
```

```
 32    FORMAT(1X, 16H1     ENTER NODES)                                 MAT00590
       WRITE(6,33)                                                      MAT00600
 33    FORMAT(1X, 15H2     ENTER ARCS)                                  MAT00610
       WRITE(6,34)                                                      MAT00620
 34    FORMAT(1X, 25H3     SET ITERATION NUMBER)                        MAT00630
       WRITE(6,35)                                                      MAT00640
 35    FORMAT(1X, 24H4     SCAN THE NET SO FAR)                         MAT00650
       WRITE(6,36)                                                      MAT00660
 36    FORMAT(1X, 12H5     RUN NET)                                     MAT00670
       WRITE(6,37)                                                      MAT00680
 37    FORMAT(1X, 25H6     ENTER RUN IDENTIFIER)                        MAT00690
       WRITE(6,38)                                                      MAT00700
 38    FORMAT(1X, 16H9     END SESSION)                                 MAT00710
       GO TO 7                                                          MAT00720
 77    WRITE(6,50)                                                      MAT00730
       WRITE(6,78)                                                      MAT00740
 78    FORMAT(1X, 33HYOU HAVE RETURNED TO MONITOR MODE)                 MAT00750
       WRITE(6,79)                                                      MAT00760
 79    FORMAT(1X, 55HSELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFMAT00770
      CORE)                                                             MAT00771
 7     WRITE(6,50)                                                      MAT00790
 50    FORMAT(2H *)                                                     MAT00800
       READ(5,1) MODTYP                                                 MAT00810
 1     FORMAT(I1)                                                       MAT00820
       IF(MODTYP.EQ.1) CALLNODIN                                        MAT00830
       IF(MODTYP.EQ.2) CALL ARCIN                                       MAT00840
       IF(MODTYP.EQ.3) CALL REPSET                                      MAT00850
       IF(MODTYP.EQ.4) CALL SCAN                                        MAT00860
       IF(MODTYP.EQ.5) CALL RUNSYS                                      MAT00870
        IF(MODTYP.EQ.6) CALL IDIN                                       MAT00880
       IF(MODTYP.EQ.9) CALL EXIT                                        MAT00890
       GO TO 77                                                         MAT00900
       END                                                              MAT00910



       SUBROUTINE ARCIN                                                 MAT00920
C                                                                       MAT00930
C      THIS ROUTINE WILL SERVE TO READ IN ARCS AND ASSOCIATED DATA      MAT00940
C                                                                       MAT00950
       COMMON/INDEV/INDV                                                MAT00960
       COMMON/PARA/NNODE,NARC                                           MAT00990
       COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                     MAT01000
       COMMON/ARC1/ ARC(500),INODE(500),ONODE(500),ITIMET(500),         MAT00970
      CTARG1(500),TARG2(500),TARG3(500),COSTC(500),COSTV(500)           MAT00971
       COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),   MAT01010
      CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),          MAT01011
      CTIMEN(100)                                                       MAT01012
       REAL NODE                                                        MAT01040
       INTEGER ONODE                                                    MAT01080
       INTEGER OARCI,OARC                                               MAT01050
       DATA RETC/4HRETU/                                                MAT01060
C                                                                       MAT01070
C                                                                       MAT01090
       IF(INDV.EQ.0) GO TO 50                                           MAT01100
       WRITE(6,60)                                                      MAT01110
 60    FORMAT(1X, 29HYOU ARE NOW IN ENTER ARC MODE)                     MAT01120
 150   WRITE(6,61)                                                      MAT01130
 61    FORMAT(2H *)                                                     MAT01140
```

```
      WRITE(6,70)                                                       MAT01150
70    FORMAT(1X, 48HENTER ARC NAME,INPUT NODE NAME,OUTPUT NODE NAME,/   MAT01160
     C  1X, 57HTIME DISTRIBUTION TYPE,TIME DISTRIBUTION ARGUMENTS 1,2,3,MAT01161
     C    /1X, 71HCONSTANT COST COEFFICIENT,COEFFICIENT OF TIME TERM IN MAT01162
     C         COST TERM,//1X, 40HPROBABILITY OF SUCCESSFUL ARC COMPLETIONMAT01163
     C)                                                                 MAT01164
      WRITE(6,71)                                                       MAT01200
71    FORMAT(1X, 23HFORMAT IS 3A4,I1,6F10.0)                            MAT01210
      WRITE(6,61)                                                       MAT01220
      WRITE(6,72)                                                       MAT01230
77     WRITE(6,61)                                                      MAT01240
72    FORMAT(1X, 38HTO RETURN TO MONITOR MODE ENTER  RETU )             MAT01250
50    READ(5,1) ANAME,AINODE,AONODE,IDIST,P1,P2,P3,C1,C2,D              MAT01260
    1 FORMAT(3A4,I1,5F10.0,F9.0)
      IF(ANAME.EQ.RETC) RETURN                                          MAT01280
      NARC=NARC+1                                                       MAT01290
      ARC(NARC)=ANAME                                                   MAT01300
      ITIMET(NARC)=IDIST                                                MAT01310
      TARG1(NARC)=P1                                                    MAT01320
      TARG2(NARC)=P2                                                    MAT01330
      TARG3(NARC)=P3                                                    MAT01340
      COSTC(NARC)=C1                                                    MAT01350
      COSTV(NARC)=C2                                                    MAT01360
```

```
      PROB(NARC)=D                                                      MAT01370
      IF(NNODE.EQ.0) GO TO 10                                           MAT01380
      DO 9 I=1,NNODE                                                    MAT01390
      ISAVE=I                                                           MAT01400
C     ISAVE SIMPLY KEEPS VALUE OF I TO USE OUT OF DO LOOP               MAT01410
      IF(AINODE.EQ.NODE(I)) GO TO 11                                    MAT01420
 9    CONTINUE                                                          MAT01430
 10      NNODE=NNODE+1                                                  MAT01440
      NODE(NNODE)=AINODE                                                MAT01450
      INODE(NARC)=NNODE                                                 MAT01460
      OARC(NNODE,1)=NARC                                                MAT01470
      OARCI(NNODE)=1                                                    MAT01480
      GO TO 12                                                          MAT01490
 11   INODE(NARC)=ISAVE                                                 MAT01500
      OARCI(ISAVE)=OARCI(ISAVE)+1                                       MAT01510
      LM=OARCI(ISAVE)                                                   MAT01520
      OARC(ISAVE,LM)=NARC                                               MAT01530
 12   CONTINUE                                                          MAT01540
      DO 20 I=1,NNODE                                                   MAT01550
      ISAVE=I                                                           MAT01560
      IF(AONODE.EQ.NODE(I)) GO TO 30                                    MAT01570
 20   CONTINUE                                                          MAT01580
C     WE HAVE A NEW NODE                                                MAT01590
      NNODE=NNODE+1                                                     MAT01600
      NODE(NNODE)=AONODE                                                MAT01610
      ONODE(NARC)=NNODE                                                 MAT01620
      IARC(NNODE,1)=NARC                                                MAT01630
      IARCI(NNODE)=1                                                    MAT01640
      GO TO 40                                                          MAT01650
 30   ONODE(NARC)=ISAVE                                                 MAT01660
      IARCI(ISAVE)=IARCI(ISAVE)+1                                       MAT01670
      LM=IARCI(ISAVE)                                                   MAT01680
      IARC(ISAVE,LM)=NARC                                               MAT01690
 40   CONTINUE                                                          MAT01700
       GO TO 77                                                         MAT01710
      END                                                               MAT01720


      SUBROUTINE NODIN                                                  MAT01730
C     THIS ROUTINE WILL READ IN NODE NAMES AND DATA                     MAT01740
C                                                                       MAT01750
      COMMON/INDEV/ INDV                                                MAT01760
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT01770
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT01771
     CTIMEN(100)                                                        MAT01772
      REAL NODE                                                         MAT01800
      DIMENSION DDUM(10)                                                MAT01810
      COMMON/PARA/ NNODE,NARC                                           MAT01820
      COMMON/TERNN/NODN(30),NODI,TIMEZ(1000),COSTZ(1000),NODEZ(1000),   MAT01830
     CNCOUNT(30)                                                        MAT01831
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT01850
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT01860
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                       MAT01861
      DIMENSION ZNAM(10),ZPROB(10),IZNAM(10),YI(10),YO(10)              MAT01880
      INTEGER OARC,OARCI                                                MAT01900
      INTEGER OTYPE                                                     MAT01930
      COMMON/TERNI/INODI(10),INODT                                      MAT01920
      DATA CZZ/4HZZZZ/                                                  MAT01890
      DATA RETC/4HRETU/                                                 MAT01910
```

```
C
C     TEST IF BATCH, IF SO SKIP GOBBELDYGOOK
C
      IF(INDV.EQ.0) GO TO 20
C     TERMINAL MODE PRINT INSTRUCTIONS
      WRITE(6,1)
 1    FORMAT(1X, 26HYOU ARE IN ENTER NODE MODE)
      WRITE(6,50)
 50   FORMAT(1X, 38HENTER NODE NAME,INPUT RULE,OUTPUT RULE)
      WRITE(6,51)
 51   FORMAT(1X, 18HFORMAT IS A4,I1,I1)
      WRITE(6,52)
 52   FORMAT(1X, 37HINPUT AND OUTPUT RULES ARE AS FOLLOWS)
      WRITE(6,53)
 53   FORMAT(10X, 11HRULE NUMBER,10X, 10HINPUT RULE,5X, 11HOUTPUT RULE)
      WRITE(6,540)
 540  FORMAT(9X,13(1H.),8X,12(1H.),13X,13(1H.))
      WRITE(6,54)
 54   FORMAT(15X,1H1,18X,  3HAND,20X,  8HALL FIRE)
      WRITE(6,55)
 55   FORMAT(15X,1H2,18X,  2HOR,21X, 10HPROB. FIRE)
      WRITE(6,56)
 56    FORMAT(15X,1H4,18X,  7HINITIAL,16X,  8HTERMINAL)
      WRITE(6,57)
 57   FORMAT(15X,1H5,18X,  3H1/1,20X,  3H1/1)
      WRITE(6,58)
 58   FORMAT(15X,1H6,18X,  7H1/1 BAR,16X,  7H1/1 BAR)
      WRITE(6,59)
 59   FORMAT(15X,1H7,18X,  9HPREFERRED,14X,  9HPREFERRED)
 20   WRITE(6,30)
 30   FORMAT(2H *)
      READ(5,40) ADUM,IDUM,IODUM
 40   FORMAT(A4,I1,I1)
      IF(ADUM.EQ.RETC) RETURN
      DO 100 I=1,NNODE
      NDUM=I
      IF(NODE(I).EQ.ADUM) GO TO 101
 100  CONTINUE
      NNODE=NNODE+1
      NDUM=NNODE
 101  ITYPE(NDUM)=IDUM
      OTYPE(NDUM)=IODUM
      IF(ITYPE(NDUM).EQ.4) GO TO 90
C     ITYPE=4 MEANS INITIAL NODE
 222  IF(ITYPE(NDUM).EQ.6) GO TO 400
C     ITYPE=6 MEANS WE HAVE A 1/1 NODE WITH A NEGATIVE INPUT
      IF(ITYPE(NDUM).EQ.5) GO TO 60
C     ITYPE= 5 MEANS A 1/1 NODE WHICH MEANS WE MUST READ IN ORDERINGS
      IF(OTYPE(NDUM).EQ.2) GO TO 70
C     OTYPE=2 MEANS PROBABILISTIC FIRINGS..WE MUST READ IN PROBABILITYS
      IF(OTYPE(NDUM).EQ.4) GO TO 80
      IF(OTYPE(NDUM).EQ.7) GO TO 600
      GO TO 20
 60   CONTINUE
C     INSERT READS HERE FOR 1/1 NODES
      WRITE(6,300)
 300  FORMAT(1X, 29HYOU HAVE INDICATED A 1/1 NODE)
      WRITE(6,301)
 301  FORMAT(1X, 52HINPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAMES
     C)
```

```
      WRITE(6,302)
  302 FORMAT(1X, 23HFORMAT IS I2,10(A4,A4   )
      WRITE(6,30)
      READ(5,303)  MM,(YI(I),YO(I),I=1,MM)
  303 FORMAT(I2,10(A4,A4))
  203 FORMAT(I2,10(A4,F6.3))
      DO 305 L=1,NARC
      DO 306 K=1,MM
      IF(ARC(L).EQ.YI(K)) IARC(NDUM,K)=L
      IF(ARC(L).EQ.YO(K)) OARC(NDUM,K)=L
  306 CONTINUE
  305 CONTINUE
      GO TO 20
   70 CONTINUE
C     INSERT READ FOR PROBABILITIES
      IF(INDV.EQ.0) GO TO 220
      WRITE(6,200)
  200 FORMAT(1X, 49HYOU HAVE INDICATED A NODE WITH STOCHASTIC OUTPUTS)
      WRITE(6,201)
  201 FORMAT(1X, 53HINPUT NUMBER OF OUTPUT ARCS NAME OF OUTPUT ARC,PROB.
     C.)
      WRITE(6,202)
  202 FORMAT(1X, 25HFORMAT IS I2,10(A4,F6.3) )
      WRITE(6,30)
  220 READ(5,203) NN,(ZNAM(I),ZPROB(I),I=1,NN)
      IF(NN.NE.OARCI(NDUM)) CALL TERM(99)
      DO 205 I=1,NN
      DO 206 J=1,NN
      JJ=J
      LM=OARC(NDUM,I)
      IF(ZNAM(J).EQ.ARC(LM)) GO TO 207
  206 CONTINUE
      CALL TERM(100)
  207 PPOA(NDUM,I)=ZPROB(JJ)
  205 CONTINUE
      GO TO 20
  80  NODI=NODI+1
      NODN(NODI)=NDUM
      GO TO 20
  90     INODT=INODT+1
      INODI(INODT)=NDUM
      GO TO 222
  400 IF(INDV.EQ.0) GO TO 420
      WRITE(6,401)
  401 FORMAT(1X, 33HYOU HAVE INDICATED A 1/1 BAR NODE)
      WRITE(6,402)
  402 FORMAT(1X, 51HINPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME)
         WRITE(6,403)
  403 FORMAT(1X, 47HINPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION)
      WRITE(6,302)
      WRITE(6,30)
  420 READ(5,303) MM,(YI(I),YO(I),I=1,MM)
      DO 405 L=1,NARC
      DO 406 K=1,MM
      IF(ARC(L).EQ.YI(K)) IARC(NDUM,K)=L
      IF(ARC(L).EQ.YO(K)) OARC(NDUM,K)=L
      IF(YI(K).EQ.CZZ) IARC(NDUM,K)=500
  406 CONTINUE
  405 CONTINUE
      IARCI(NDUM)=IARCI(NDUM)+1
```

```
      GO TO 20                                                          MAT03120
  600 IF(INDV.EQ.0) GO TO 666                                           MAT03130
      WRITE(6,601)                                                      MAT03140
  601 FORMAT(1X, 35HYOU HAVE INDICATED A PREFERRED NODE)                MAT03150
      WRITE(6,302)                                                      MAT03160
      WRITE(6,602)                                                      MAT03170
  602 FORMAT(1X, 43HI/O ARC PAIRS SHOULD BE IN PREFERENCE ORDER)        MAT03180
      WRITE(6,403)                                                      MAT03190
      WRITE(6,30)                                                       MAT03200
  666 READ(5,303) MM,(YI(I),YO(I),I=1,MM)                               MAT03210
      DO 605 L=1,NARC                                                   MAT03220
      DO 606 K=1,MM                                                     MAT03230
      IF(ARC(L).EQ.YI(K)) IARC(NDUM,K)=L                                MAT03240
      IF(ARC(L).EQ.YO(K)) OARC(NDUM,K)=L                                MAT03250
      IF(YI(K).EQ.CZZ) IARC(NDUM,K)=500                                 MAT03260
  606 CONTINUE                                                          MAT03270
  605 CONTINUE                                                          MAT03280
      IARCI(NDUM)=IARCI(NDUM)+1                                         MAT03290
      GO TO 20                                                          MAT03300
      END                                                               MAT03310


       SUBROUTINE IDIN                                                  MAT03320
       COMMON/IDD/ RUNID(20)                                            MAT03330
       WRITE(6,1)                                                       MAT03340
    1 FORMAT(1X, 47HENTER A RUN IDENTIFIER OF 80 CHARACTERS OR LESS)    MAT03350
       READ(5,2) RUNID                                                  MAT03360
    2 FORMAT(20A4)                                                      MAT03370
       RETURN                                                           MAT03380
       END                                                              MAT03390


      SUBROUTINE REPSET                                                 MAT03400
      COMMON/ITERA/ITER                                                 MAT03410
      COMMON/INDEV/INDV                                                 MAT03420
C     TEST IF IN BATCH MODE,IF SO SKIP GOBBELDY GOOK                    MAT03430
      IF(INDV.EQ.0) GO TO 4                                             MAT03440
      WRITE(6,1)                                                        MAT03450
    1 FORMAT(1X, 40HYOU CAN NOW SET THE NUMBER OF ITERATIONS)           MAT03460
      WRITE(6,2)                                                        MAT03470
    2 FORMAT(1X, 42HENTER A 5 POSITION INTEGER, RIGHT ADJUSTED)         MAT03480
      WRITE(6,3)                                                        MAT03490
    3 FORMAT(1X,1H*)                                                    MAT03500
    4 READ(5,5) ITER                                                    MAT03510
    5 FORMAT(I5)                                                        MAT03520
      RETURN                                                            MAT03530
      END                                                               MAT03540


      SUBROUTINE SCAN                                                   MAT03550
C     THIS ROUTINE WILL PRINT OUT THE NET TO DATE                       MAT03560
      COMMON/RANC/ IXX                                                  MAT03570
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT03580
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT03590
```

```
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                        MAT03591
      COMMON/PARA/ NNODE,NARC                                            MAT03610
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),     MAT03620
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),            MAT03621
     CTIMEN(100)                                                         MAT03622
      REAL NODE                                                          MAT03650
      INTEGER ONODE,OTYPE                                                MAT03660
C                                                                        MAT03670
      WRITE(6,1)                                                         MAT03680
    1 FORMAT(1X,47H ARC      INPUT NODE      OUTPUT NODE     TIME DIST   ,
     156H    ARG1        ARG2        ARG3           COST           PROB   )
      DO 2 I=1,NARC                                                      MAT03710
      LM=INODE(I)                                                        MAT03720
      LN=ONODE(I)                                                        MAT03730
      WRITE(6,3)ARC(I),NODE(LM),NODE(LN),ITIMET(I),
     1TARG1(I),TARG2(I),TARG3(I),COSTC(I),COSTV(I),PROB(I)
    3 FORMAT(2X,A4,6X,A4,11X,A4,11X,I1,5X,F7.2,3(3X,F7.2),
     1 2H +,F7.2,3X,F6.3)
  2   CONTINUE                                                           MAT03780
C                                                                        MAT03790
C     NOW WRITE OUT ARCS AND ASSOCIATED DATA                             MAT03800
C                                                                        MAT03810
      WRITE(6,50)                                                        MAT03820
 50   FORMAT(1H )                                                        MAT03830
      WRITE(6,51)                                                        MAT03840
 51   FORMAT(1X, 53H    NODE           NO. OF INPUT ARCS    NO. OF OUTPUT ARCMAT03850
     CS, 29H     INPUT TYPE      OUTPUT TYPE)                            MAT03851
      DO 60 I=1,NNODE                                                    MAT03870
      WRITE(6,61) NODE(I),IARCI(I),OARCI(I),ITYPE(I),OTYPE(I)            MAT03880
 61   FORMAT(3X,A4,15X,I2,15X,I2,20X,I2,15X,I2)                          MAT03890
 60   CONTINUE                                                           MAT03900
      RETURN                                                             MAT03910
      END                                                                MAT03920



      SUBROUTINE RUNSYS                                                  MAT03930
C     THIS ROUTINE CONTROLS THE RUNNING OF A NET                         MAT03940
C                                                                        MAT03950
       COMMON/IRRZ/IREPIT                                                MAT03960
      COMMON/KKIND/KIND                                                  MAT03970
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT03980
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                        MAT03981
      COMMON/ITERA/ITER                                                  MAT04000
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                       MAT04010
      COMMON/RUNER/ ITERMS                                               MAT04020
      COMMON/MINT/SMTIM,INSM                                             MAT04030
      COMMON/TERNN/NODN(30),NODI,TIMEZ(1000),COSTZ(1000),NODEZ(1000),    MAT04040
     CNCOUNT(30)                                                         MAT04041
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),     MAT04060
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),            MAT04061
     CTIMEN(100)                                                         MAT04062
      COMMON/PARA/NNODE,NARC                                             MAT04090
      COMMON/TERNI/ INODI(10),INODT                                      MAT04100
      INTEGER OARC,OARCI,OTYPE                                           MAT04110
C                                                                        MAT04120
C     SET NUMBER OF ITERATIONS SO FAR TO 0                               MAT04130
      IREPIT=0                                                           MAT04140
C                                                                        MAT04150
```

```
 20    SMTIM=999999.0                                                   MAT04160
C      ABOVE INITIALIZES SMALLEST TERMINAL FINISH TIME TO A LARGE NUMBER MAT04170
C                                                                       MAT04180
C      THE FOLLOWING CODE SETS THE INDICATER OF WHETHER OR NOT          MAT04190
C      A TERMINAL NODE HAS BEEN FILLED TO INDICATE NO.                  MAT04200
C                                                                       MAT04210
       ITERMS=0                                                         MAT04220
C      NOW FIRE INITIAL NODES                                           MAT04230
C                                                                       MAT04240
       DO 10 I=1,INODT                                                  MAT04250
       LM=INODI(I)                                                      MAT04260
       TIMEN(LM)=0.                                                     MAT04270
       IF(OTYPE(LM).EQ.1) CALL ALLFIR(LM)                               MAT04280
       IF(OTYPE(LM).EQ.2) CALL PROFIR(LM)                               MAT04290
 10    CONTINUE                                                         MAT04300
 1     CALL ARCCHK                                                      MAT04310
       CALL NODCHK                                                      MAT04320
       IF(KIND.EQ.0) GO TO 60                                           MAT04330
       IF(ITERMS.EQ.0) GO TO 1                                          MAT04340
       CALL ENDIT(KEY)                                                  MAT04350
       IF(KEY.EQ.0) GO TO 1                                             MAT04360
 61    CALL PTERM                                                       MAT04370
       IREPIT=IREPIT+1                                                  MAT04380
       IF(IREPIT.EQ.ITER) CALL SGRAPH                                   MAT04390
       IF(IREPIT.EQ.ITER) RETURN                                        MAT04400
       DO 40 I=1,NARC                                                   MAT04410
       TIME(I)=0.                                                       MAT04420
       ISTAT(I)=0                                                       MAT04430
 40    CONTINUE                                                         MAT04440
       DO 50 I=1,NNODE                                                  MAT04450
       TIMEN(I)=0.                                                      MAT04460
 50    CONTINUE                                                         MAT04470
       GO TO 20                                                         MAT04480
C      COME HERE IF NO NODES HAVE FIRED                                 MAT04490
 60    IF(ITERMS.EQ.0) CALL TERM(3030)                                  MAT04500
C      DO ABOVE IF NO TERMINAL NODES HAVE BEEN FILLED                   MAT04510
       GO TO 61                                                         MAT04520
C      DO ABOVE IF THERE IS A TERMINAL NODE-END ITERATION               MAT04530
       END                                                              MAT04540



       SUBROUTINE NODCHK                                                MAT04550
C      SEE WHAT NODES ARE READY TO FIRE,FIRE THOSE THAT                 MAT04560
C      ARE READY                                                        MAT04570
C                                                                       MAT04580
       COMMON/PARA/NNODE,NARC                                           MAT04590
       COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),   MAT04600
      CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),          MAT04601
      CTIMEN(100)                                                       MAT04602
       COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                     MAT04630
       INTEGER OTYPE,OARCI,OARC                                         MAT04640
       COMMON /KKIND/KIND                                               MAT04650
       REAL NODE                                                        MAT04660
       KIND=0                                                           MAT04670
       DO 1 I=1,NNODE                                                   MAT04680
       IL=I                                                             MAT04690
       J=0                                                              MAT04700
       IF(ITYPE(I).NE.1) GO TO 40                                       MAT04710
```

```
      CALL ANDTST(IL,J)
      GO TO 100
 40   IF(ITYPE(I).NE.2) GO TO 41
      CALL ORTST(IL,J)
      GO TO 100
 41   CONTINUE
      IF(ITYPE(I).EQ.5) GO TO 52
      IF(ITYPE(I).EQ.6) GO TO 53
       IF(ITYPE(I).EQ.7) GO TO 54
 100  IF(J.EQ.0) GO TO 1
      KIND=KIND+1
C
C     NOW ZERO OUT INPUT ARCS
C
      IRK=IARCI(I)
      DO 22 IJ=1,IRK
      LM=IARC(I,IJ)
       IF(ISTAT(LM).EQ.0) GO TO 22
      ISTAT(LM)=4
 22   CONTINUE
      IF(OTYPE(I).NE.1) GO TO 50
      CALL ALLFIR(I)
      GO TO 1
 50   IF(OTYPE(I).NE.2) GO TO 51
      CALL PREFIR(I)
      GO TO 1
 51   IF(OTYPE(I).NE.4) GO TO 52
      CALL ITALL(I)
      GO TO 1
 52   IF(OTYPE(I).NE.5) CALL TERM(1210)
      CALL ONEONE(IL,J)
      KIND=KIND+J
      GO TO 1
 53   IF(OTYPE(I).NE.6) CALL TERM(1211)
      CALL ONEBAR(IL,J)
      KIND=KIND+J
      GO TO 1
 54   IF(OTYPE(I).NE.7) CALL TERM(1212)
      CALL PREFER(IL,J)
      KIND=KIND+J
 1    CONTINUE
      RETURN
      END


      SUBROUTINE ANDTST(I,J)
C     THIS ROUTINE TESTS AND  NODES
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
     CTIMEN(100)
      REAL NODE
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)
      II=IARCI(I)
      TIM=0.
      DO 1 K=1,II
      KK=IARC(I,K)
      IF(ISTAT(KK).NE.2) GO TO 2
      IF(TIME(KK).GT.TIM) TIM=TIME(KK)
```

```
1     CONTINUE                                                          MAT05280
      TIMEN(I)=TIM                                                      MAT05290
      J=1                                                               MAT05300
      RETURN                                                            MAT05310
2     J=0                                                               MAT05320
      RETURN                                                            MAT05330
      END                                                               MAT05340



      SUBROUTINE ORTST(I,J)                                             MAT05350
C     THIS ROUTINE TEST  OR  NODES                                      MAT05360
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT05370
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT05371
     CTIMEN(100)                                                        MAT05372
      REAL NODE                                                         MAT05400
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT05410
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT05420
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                       MAT05421
      II=IARCI(I)                                                       MAT05440
      TIM=1000000.                                                      MAT05450
      J=0                                                               MAT05460
      DO 1 K=1,II                                                       MAT05470
      KK=IARC(I,K)                                                      MAT05480
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAT05490
      J=1                                                               MAT05500
      IF(TIME(KK).LT.TIM) TIM=TIME(KK)                                  MAT05510
1     CONTINUE                                                          MAT05520
      IF(J.EQ.0) RETURN                                                 MAT05530
      TIMEN(I)=TIM                                                      MAT05540
      RETURN                                                            MAT05550
      END                                                               MAT05560



      SUBROUTINE PROFIR(I)                                              MAT05570
C     FIRE NODE I USING STOCHASTIC CONSIDERATIONS                       MAT05580
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT05590
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT05600
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT05601
     CTIMEN(100)                                                        MAT05602
      REAL NODE                                                         MAT05630
      COMMON/RANC/ IXX                                                  MAT05640
      INTEGER OARC,OARCI,OTYPE                                          MAT05650
C                                                                       MAT05660
      IXXX=IXX                                                          MAT05670
      CALL RANDU(IXXX,IXY,RVAL)                                         MAT05680
      IXX=IXY                                                           MAT05690
      AHIGH=0.                                                          MAT05700
      II=OARCI(I)                                                       MAT05710
      DO 1 K=1,II                                                       MAT05720
      KK=OARC(I,K)                                                      MAT05730
      ALOW=AHIGH                                                        MAT05740
      AHIGH=ALOW+PPOA(I,K)                                              MAT05750
      IF((RVAL.GE.ALOW).AND.(RVAL.LE.AHIGH)) GO TO 2                    MAT05760
1     CONTINUE                                                          MAT05770
      CALL TERM(1)                                                      MAT05780
2     ISTAT(KK)=1                                                       MAT05790
```

```
      RETURN                                                            MAT05800
      END                                                               MAT05810
```

```
      SUBROUTINE ALLFIR(I)                                              MAT05820
C     FIRE ALL OUTPUT ARCS                                              MAT05830
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT05840
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT05850
     CIARCI(100), OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),          MAT05851
     CTIMEN(100)                                                        MAT05852
      REAL NODE                                                         MAT05880
      INTEGER OARCI,OARC                                                MAT05890
      II=OARCI(I)                                                       MAT05900
      DO 1 K=1,II                                                       MAT05910
      KK=OARC(I,K)                                                      MAT05920
      ISTAT(KK)=1                                                       MAT05930
 1    CONTINUE                                                          MAT05940
      RETURN                                                            MAT05950
      END                                                               MAT05960
```

```
       SUBROUTINE ONEONE(ILK,J)                                         MAT05970
      COMMON/PARA/NNODE,NARC                                            MAT05980
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT05990
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT05991
     CTIMEN(100)                                                        MAT05992
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT06020
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                       MAT06021
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT06040
      INTEGER OTYPE,OARCI,OARC                                          MAT06050
      REAL NODE                                                         MAT06060
      J=0                                                               MAT06070
      II=IARCI(ILK)                                                     MAT06080
      TIM=1000000.                                                      MAT06090
      DO 1 K=1,II                                                       MAT06100
      KK=IARC(ILK,K)                                                    MAT06110
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAT06120
      IF(TIME(KK).GE.TIM) GO TO 1                                       MAT06130
      TIM=TIME(KK)                                                      MAT06140
      J=K                                                               MAT06150
 1    CONTINUE                                                          MAT06160
      IF(J.EQ.0) RETURN                                                 MAT06170
C     ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC                         MAT06180
      DO 2 K=1,II                                                       MAT06190
      LM=IARC(ILK,K)                                                    MAT06200
       IF(ISTAT(LM).EQ.0) GO TO 2                                       MAT06210
       ISTAT(LM)=4                                                      MAT06220
 2    CONTINUE                                                          MAT06230
      LM=OARC(ILK,J)                                                    MAT06240
      ISTAT(LM)=1                                                       MAT06250
      TIMEN(ILK)=TIM                                                    MAT06260
      RETURN                                                            MAT06270
      END                                                               MAT06280
```

```
      SUBROUTINE ONEBAR(ILK,J)                                          MAT06290
      COMMON/PARA/NNODE,NARC                                            MAT06300
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT06310
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT06311
     CTIMEN(100)                                                        MAT06312
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT06340
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                       MAT06341
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT06360
      INTEGER OTYPE,OARCI,OARC                                          MAT06370
      REAL NODE                                                         MAT06380
      J=0                                                               MAT06390
      II=IARCI(ILK)                                                     MAT06400
      TIM=1000000.                                                      MAT06410
      DO 1 K=1,II                                                       MAT06420
      KK=IARC(ILK,K)                                                    MAT06430
      IF(KK.EQ.500) IBAR=K                                              MAT06440
      IF(KK.EQ.500) GO TO 1                                             MAT06450
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAT06460
      IF(TIME(KK).GE.TIM) GO TO 1                                       MAT06470
      TIM=TIME(KK)                                                      MAT06480
      J=K                                                               MAT06490
    1 CONTINUE                                                          MAT06500
      IF(J.EQ.0) GO TO 5                                                MAT06510
C     ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC                         MAT06520
      DO 2 K=1,II                                                       MAT06530
      LM=IARC(ILK,K)                                                    MAT06540
       IF(ISTAT(LM).EQ.0) GO TO 2                                       MAT06550
       ISTAT(LM)=4                                                      MAT06560
    2 CONTINUE                                                          MAT06570
      LM=OARC(ILK,J)                                                    MAT06580
      ISTAT(LM)=1                                                       MAT06590
      TIMEN(ILK)=TIM                                                    MAT06600
      RETURN                                                            MAT06610
    5  TIM=0.                                                           MAT06620
      DO 6 K=1,II                                                       MAT06630
      KK=IARC(ILK,K)                                                    MAT06640
      IF(KK.EQ.500) GO TO 6                                             MAT06650
      IF(ISTAT(KK).NE.3) GO TO 7                                        MAT06660
       IF(TIME(KK).LE.TIM) GO TO 6                                      MAT06670
      TIM=TIME(KK)                                                      MAT06680
    6 CONTINUE                                                          MAT06690
      LM=OARC(ILK,IBAR)                                                 MAT06700
      ISTAT(LM)=1                                                       MAT06710
       J=1                                                              MAT06720
      DO 12 K=1,II                                                      MAT06730
      LM=IARC(ILK,K)                                                    MAT06740
       ISTAT(LM)=4                                                      MAT06750
   12 CONTINUE                                                          MAT06760
      TIMEN(ILK)=TIM                                                    MAT06770
    7 CONTINUE                                                          MAT06780
      RETURN                                                            MAT06790
      END                                                               MAT06800


      SUBROUTINE PREFER(ILK,J)                                          MAT06810
      COMMON/PARA/NNODE,NARC                                            MAT06820
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT06830
```

```
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
     CTIMEN(100)
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)
      INTEGER OTYPE,OARCI,OARC
      REAL NODE
       J=0
       II=IARCI(ILK)
       TIM=0.
       DO 1 K=1,II
       KK=IARC(ILK,K)
       IF(KK.EQ.500) GO TO 1
       IF(ISTAT(KK).EQ.0) GO TO 30
       IF(ISTAT(KK).EQ.1) GO TO 30
       IF(ISTAT(KK).EQ.4) GO TO 30
       IF(TIME(KK).GT.TIM) TIM=TIME(KK)
 1     CONTINUE
       J=1
C     IF WE GET HERE THE NODE WILL BE FIRED
C      FIRE FIRST ARC PAIR WITH 2 STATUS, IF THERE IS ONE
      III=II-1
       DO 2 K=1,III
       KK=IARC(ILK,K)
       KKK=K
       IF(ISTAT(KK).EQ.2) GO TO 4
 2     CONTINUE
C     IF WE GET HERE FIRE BAR ARC
       LM=OARC(ILK,II)
       GO TO 5
C      FIRE THE KKK ARC
 4     LM=OARC(ILK,KKK)
      LN=IARC(ILK,KKK)
       TIM=TIME(LN)
 5     ISTAT(LM)=1
       DO 40 K=1,II
       LM=IARC(ILK,K)
       ISTAT(LM)=4
 40    CONTINUE
 30    CONTINUE
        TIMEN(ILK)=TIM
       RETURN
       END



      SUBROUTINE ARCCHK
C     IF INITIATED8CHECK PROBABILITY OF COMPLETION
C     IF COMPLETED8CALCULATE TIME AND COST
C
      COMMON/RANC/ IXX
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)
      COMMON/PARA/ NNODE,NARC
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
     CTIMEN(100)
      REAL NODE
```

```
      DO 1 I=1,NARC                                                     MAT07400
      IF(ISTAT(I).NE.1) GO TO 1                                         MAT07410
      IXXX=IXX                                                          MAT07420
      A=TARG1(I)                                                        MAT07430
      B=TARG2(I)                                                        MAT07440
      C=TARG3(I)                                                        MAT07450
      IF(ITIMET(I).EQ.1) CALL GAUSS(IXXX,B,A,TVAL)                      MAT07460
      IF(ITIMET(I).EQ.2)CALL TRIANG(IXXX,A,B,C,TVAL)                    MAT07470
      IF(ITIMET(I) .EQ. 3) CALL UNIFRM(IXXX,A,B,C,TVAL)
      IXX=IXXX                                                          MAT07480
      LM=INODE(I)                                                       MAT07490
      TIME(I)=TVAL+TIMEN(LM)                                            MAT07500
      IF(PROB(I).EQ.1) GO TO 2                                          MAT07510
      IXXX=IXX                                                          MAT07520
      CALL RANDU(IXXX,IXYY,RVAL)                                        MAT07530
      IXX=IXYY                                                          MAT07540
      IF(RVAL.LE.PROB(I)) GO TO 2                                       MAT07550
      ISTAT(I)=3                                                        MAT07560
      GO TO 1                                                           MAT07570
    2 ISTAT(I)=2                                                        MAT07580
    1 CONTINUE                                                          MAT07590
      RETURN                                                            MAT07600
      END                                                               MAT07610


      SUBROUTINE ITALL(II)                                              MAT07620
C                                                                       MAT07630
C     THIS ROUTINE WILL HANDLE A TERMINAL NODE BEING FILLED             MAT07640
C     IT WILL SEE IF THE TIME IS SMALLER THAN ANY OTHER TERMINAL        MAT07650
C     NODE TIME AND IF SO SWAP TIME AND COST INDICATORS                 MAT07660
C                                                                       MAT07670
      COMMON/MINT/SMTIM,INSM                                            MAT07680
      COMMON/RUNER/ ITERMS                                              MAT07690
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT07700
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT07701
     CTIMEN(100)                                                        MAT07702
      ITERMS=1                                                          MAT07730
C     ABOVE INDICATES A TERMINAL NODE HAS BEEN FILLED                   MAT07740
      IF(TIMEN(II).GE.SMTIM) RETURN                                     MAT07750
      SMTIM=TIMEN(II)                                                   MAT07760
      INSM=II                                                           MAT07770
      RETURN                                                            MAT07780
      END                                                               MAT07790


      SUBROUTINE ENDIT(KEY)                                             MAT07800
C                                                                       MAT07810
C     THIS ROUTINE CHECKS TO SEE IF THERE ARE COMPLETED ARCS WITH TIMES MAT07820
C     SMALLER THAN THE SMALLEST TERMINAL NODE                           MAT07830
C     IF SO SET KEY=0, IF NOT SET KEY=1 AND TERMINATE THIS RUN          MAT07840
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT07850
      COMMON/MINT/ SMTIM,INSM                                           MAT07860
      COMMON/PARA/NNODE,NARC                                            MAT07870
      DO 1 I=1,NARC                                                     MAT07880
      IF(ISTAT(I).EQ.0) GO TO 1                                         MAT07890
      IF(ISTAT(I).EQ.3) GO TO 1                                         MAT07900
```

```
      IF(ISTAT(I).EQ.4) GO TO 1                                         MAT07910
      IF(TIME(I).LT.SMTIM) GO TO 2                                      MAT07920
    1 CONTINUE                                                          MAT07930
      KEY=1                                                             MAT07940
      RETURN                                                            MAT07950
    2 KEY=0                                                             MAT07960
      RETURN                                                            MAT07970
      END                                                               MAT07980



      SUBROUTINE PTERM                                                  MAT07990
       COMMON/PARA/ NNODE,NARC                                          MAT08000
       COMMON/IRRZ/IREPIT                                               MAT08010
      COMMON/MINT/ SMTIM,INSM                                           MAT08020
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT08030
     CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT08031
     CTIMEN(100)                                                        MAT08032
      COMMON/TERNN/NODN(30),NODI,TIMEZ(1000),COSTZ(1000),NODEZ(1000),   MAT08060
     CNCOUNT(30)                                                        MAT08061
      COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),MAT08080
     CTARG2(500),TARG3(500),COSTC(500),COSTV(500)                       MAT08081
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT08100
      COMMON/TERNI/ INODI(10),INODT                                     MAT08110
C                                                                       MAT08120
      DO 1 I=1,NODI                                                     MAT08130
      JJ=I                                                              MAT08140
      IF(NODN(I).EQ.INSM) GO TO 2                                       MAT08150
    1 CONTINUE                                                          MAT08160
      CALL TERM(69)                                                     MAT08170
    2 NCOUNT(JJ)=NCOUNT(JJ)+1                                           MAT08180
      LM=NCOUNT(JJ)                                                     MAT08190
       IRR=IREPIT+1                                                     MAT08200
       TIMEZ(IRR)=SMTIM                                                 MAT08210
       NODEZ(IRR)=INSM                                                  MAT08220
       COSTZ(IRR)=0.                                                    MAT08230
       DO 3 I=1,NARC                                                    MAT08240
       LM=INODE(I)                                                      MAT08250
       IF(ISTAT(I).EQ.0) GO TO 3                                        MAT08260
       IF(TIME(I).GT.SMTIM) GO TO 4                                     MAT08270
       COSTZ(IRR)=COSTZ(IRR)+COSTC(I)+COSTV(I)*(TIME(I)-TIMEN(LM))      MAT08280
       GO TO 3                                                          MAT08290
    4  IF(TIMEN(LM).GT.SMTIM) GO TO 3                                   MAT08300
       COSTZ(IRR)=COSTZ(IRR)+COSTC(I)+COSTV(I)*(SMTIM-TIMEN(LM))        MAT08310
    3 CONTINUE                                                          MAT08320
      RETURN                                                            MAT08330
      END                                                               MAT08340


      SUBROUTINE TERM(I)                                                MAT08350
C                                                                       MAT08360
C     THIS ROUTINE WIILL ACT AS AN ERROR TERMINATOR                     MAT08370
      COMMON/ARC1/ ARC(500),INODE(500),ONODE(500),ITIMET(500),          MAT00970
     CTARG1(500),TARG2(500),TARG3(500),COSTC(500),COSTV(500)            MAT00971
      COMMON/PARA/NNODE,NARC                                            MAT00990
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT01000
      WRITE(6,1) I                                                      MAT08380
    1 FORMAT(1X, 31HEXECUTION TERMINATED FOR REASON,I5)                 MAT08390
```

```
      IF(I .EQ. 3030) GO TO 2
      CALL EXIT                                                         MAT08400
    2 WRITE(6,3)
    3 FORMAT( 25H ARCS ACTIVATED THUS FAR   )
      DO 4 J = 1,NARC
      IF(ISTAT(J) .NE. 0)WRITE(6,5) ARC(J)
    5 FORMAT(1X,A4)
    4 CONTINUE
      CALL EXIT
      END                                                               MAT08420


      SUBROUTINE RANDU(IX,IY,YFL)                                       MAT08430
      IF(IX .NE. 65539) GO TO 10
      CALL RANSET (Y)
   10 CONTINUE
      YFL = RANF(Y)
      YFL = ABS(YFL)
      IY = 1
      RETURN                                                            MAT08490
      END                                                               MAT08500


      SUBROUTINE GAUSS(IX,S,AM,V)                                       MAT08510
      A=0.0                                                             MAT08520
      DO 50 I=1,12                                                      MAT08530
      CALL RANDU(IX,IY,Y)                                               MAT08540
      IX=IY                                                             MAT08550
 50   A=A+Y                                                             MAT08560
      V=(A-6.0)*S+AM                                                    MAT08570
      RETURN                                                            MAT08580
      END                                                               MAT08590


      SUBROUTINE TRIANG(IXT,A,B,C,X)                                    MAT08600
C                                                                       MAT08610
C     THIS ROUTINE WILL CALCULATE RANDUM TRIANGULARLY DISTRIBUTED       MAT08620
C     VARIABLES                                                         MAT08630
       IF(C.EQ.A) GO TO 1                                               MAT08640
       IF(B.EQ.A) AM=0.                                                 MAT08650
       IF(B.EQ.A) GO TO 2                                               MAT08660
      AM=(B-A)/(C-A)                                                    MAT08670
 2     CONTINUE                                                         MAT08680
      CALL RANDU(IXT,IXY,VAL)                                           MAT08690
      IXT=IXY                                                           MAT08700
      IF(VAL.LE.AM) XI=SQRT(AM*VAL)                                     MAT08710
      IF(VAL.GT.AM) XI=1.-SQRT(1.-AM-VAL+AM*VAL)                        MAT08720
      X=A+XI*(C-A)                                                      MAT08730
      RETURN                                                            MAT08740
 1     X=A                                                              MAT08750
       RETURN                                                           MAT08760
      END                                                               MAT08770


      SUBROUTINE SGRAPH                                                 MAT08780
```

```
      DIMENSION DOT(120)                                                MAT08800
     DIMENSION DARR(10)                                                 MAT08880
      DIMENSION ARRR(5000)                                              MAT08900
      DIMENSION ANN(10)                                                 MAT08910
      DIMENSION NBLIP(30)                                               MAT08940
      DIMENSION AVAL(10)                                                MAT08950
     COMMON/TERNN/NODN(30),NODI,TIMEZ(1000),COSTZ(1000),NODEZ(1000),    MAT08820
    CNCOUNT(30)                                                         MAT08821
     REAL NODE                                                          MAT08840
     COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),     MAT08850
    CIARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),            MAT08851
    CTIMEN(100)                                                         MAT08852
     COMMON/ITERA/ ITER                                                 MAT08890
      COMMON/IDD/ RUNID(20)                                             MAT08970
      DATA AVAL/.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0/                         MAT08960
      DATA DOT/4H..../                                                  MAT08930
      DATA ANN/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/                 MAT08920
      DATA DOT/120*1H=/                                                 MAT08810
      DATA TT/1HI/                                                      MAT08790
C                                                                       MAT08980
C    FIRST GENERATE INDIVIDUAL GRAPHS BY NODE                           MAT08990
C                                                                       MAT09000
C                                                                       MAT09010
      DO 1 J=1,NODI                                                     MAT09020
      LM=NODN(J)                                                        MAT09030
      LL=NCOUNT(J)                                                      MAT09040
      DO 3 IZ=1,2                                                       MAT09050
      LLK=0                                                             MAT09060
      DO 2 K=1,ITER                                                     MAT09070
      IF(NODEZ(K).NE.LM) GO TO 2                                        MAT09080
      LLK=LLK+1                                                         MAT09090
      IF(IZ.EQ.1) ARRR(LLK)=TIMEZ(K)                                    MAT09100
      IF(IZ.EQ.2) ARRR(LLK)=COSTZ(K)                                    MAT09110
 2    CONTINUE                                                          MAT09120
      IF(LLK.NE.LL) WRITE(6,109) LLK,LL                                 MAT09130
 109  FORMAT(1X, 25HVALUES OF LLK AND LL ARE ,2I5)                      MAT09140
      IF(LLK.NE.LL) CALL TERM(444)                                      MAT09150
      IF(LLK.EQ.0) GO TO 3                                              MAT09160
      CALL GRAPH(ARRR,LL)                                               MAT09170
      WRITE(6,10)                                                       MAT09180
      WRITE(6,10)                                                       MAT09190
      IF(IZ.EQ.1) WRITE(6,11) NODE(LM)                                  MAT09200
      IF(IZ.EQ.2) WRITE(6,12) NODE(LM)                                  MAT09210
      WRITE(6,10)                                                       MAT09220
      WRITE(6,25) RUNID                                                 MAT09230
 25    FORMAT(20X,25A4)                                                 MAT09240
 10  FORMAT(1H )                                                        MAT09250
 11  FORMAT(20X, 44HGRAPH OF COMPLETION TIMES FOR TERMINAL NODE ,A4)    MAT09260
 12  FORMAT(20X, 44HGRAPH OF COMPLETION COSTS FOR TERMINAL NODE ,A4)    MAT09270
 3    CONTINUE                                                          MAT09280
 1   CONTINUE                                                           MAT09290
      CALL GRAPH(TIMEZ,ITER)                                            MAT09300
      WRITE(6,10)                                                       MAT09310
      WRITE(6,10)                                                       MAT09320
      WRITE(6,65)                                                       MAT09330
 65   FORMAT(20X, 39HGRAPH OF COMPLETION TIMES FOR ALL NODES)           MAT09340
      WRITE(6,10)                                                       MAT09350
      WRITE(6,25) RUNID                                                 MAT09360
      CALL GRAPH(COSTZ,ITER)                                            MAT09370
      WRITE(6,10)                                                       MAT09380
```

```
      WRITE(6,10)                                                       MAT09390
      WRITE(6,66)                                                       MAT09400
66    FORMAT(20X, 39HGRAPH OF COMPLETION COSTS FOR ALL NODES)           MAT09410
      WRITE(6,10)                                                       MAT09420
      WRITE(6,25) RUNID                                                 MAT09430
      WRITE(6,111)                                                      MAT09440
111   FORMAT(1H1)                                                       MAT09450
      DO 200 JJ=1,NODI                                                  MAT09460
      NCC=NCOUNT(JJ)                                                    MAT09470
     ROGE = FLOAT(NCC)*1000.0/FLOAT(ITER)
     DOG = ROGE - FLOAT( IFIX(ROGE)/1)
     IF(DOG .LT. .5) GO TO 80
     ROGE = ROGE + 1.0
  80 CONTINUE
 200 NBLIP(JJ) = ROGE
      DO 201 JJ=1,NODI                                                  MAT09490
     JZ = NBLIP(JJ)/100
     JY = (NBLIP(JJ) - JZ*100)/10
     JX = (NBLIP(JJ) - JZ*100 - JY*10)
      IF(JZ.EQ.0) JZ=10                                                 MAT09520
      IF(JY.EQ.0) JY=10                                                 MAT09530
     IF(JX .EQ. 0) JX = 10
      KK=NODN(JJ)                                                       MAT09540
      LL=NBLIP(JJ)                                                      MAT09550
      IF(LL.EQ.0) GO TO 201                                             MAT09560
     LL = LL/10
     IF(LL .EQ. 0) LL = 1
      WRITE(6,202) (DOT(K),K=1,LL),TT                                   MAT09570
     WRITE(6,203)NODE(KK),(DOT(K),K=1,LL),TT,DDT,ANN(JZ),ANN(JY),ANN(JX
    1)
      WRITE(6,204)                                                      MAT09590
      WRITE(6,204)                                                      MAT09600
202   FORMAT(11X,1HI,120A1)                                             MAT09610
203   FORMAT(6X,A4,1X,1HI,120A1)                                        MAT09620
204   FORMAT(11X,1HI)                                                   MAT09630
201   CONTINUE                                                          MAT09640
      WRITE(6,13)                                                       MAT09650
13    FORMAT(12X,10(10H---------I))                                     MAT09660
      WRITE(6,14) (AVAL(I),I=1,10)                                      MAT09670
14    FORMAT(13X,10(7X,F3.1))                                           MAT09680
      WRITE(6,10)                                                       MAT09690
      WRITE(6,10)                                                       MAT09700
      WRITE(6,15)                                                       MAT09710
15    FORMAT(20X, 27HGRAPH OF NODE PROBABILITIES)                       MAT09720
       WRITE(6,10)                                                      MAT09730
      WRITE(6,25) RUNID                                                 MAT09740
      WRITE(6,111)                                                      MAT09750
     RETURN                                                             MAT09760
     END                                                                MAT09770


     SUBROUTINE GRAPH(ARR,LIM)                                          MAT09790
     DIMENSION ANN(10)                                                  MAT09810
     DIMENSION ARR(LIM)                                                 MAT09840
     DIMENSION CAT(50)                                                  MAT09850
     DIMENSION ICAT(50),NUM(60),DOT(120),AVAL(10)                       MAT09870
     DATA DOT/120*1H=/                                                  MAT09890
     DATA AVAL/.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0/                          MAT09880
     DATA TERM/1HI/                                                     MAT09860
```

```
      DATA BLANK/1H /                                                   MAT09830
       DATA ANN/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/                MAT09820
       DATA DDT/4H..../                                                 MAT09800
      BIG=0.                                                            MAT09900
      SMALL=10000000.0                                                  MAT09910
      SSUM=0.                                                           MAT09920
      DO 1 I=1,LIM                                                      MAT09930
      SSUM=SSUM+ARR(I)                                                  MAT09940
      IF(ARR(I).GT.BIG) BIG=ARR(I)                                      MAT09950
      IF(ARR(I).LT.SMALL) SMALL=ARR(I)                                  MAT09960
1     CONTINUE                                                          MAT09970
      RANGE=BIG-SMALL                                                   MAT09980
      IF(RANGE.EQ.0) GO TO 900                                          MAT09990
      AINT=RANGE/25.                                                    MAT10000
      DO 4 K=1,50                                                       MAT10010
4     ICAT(K)=0                                                         MAT10020
      DO 2 J=1,LIM                                                      MAT10030
      AHIGH=SMALL                                                       MAT10040
      DO 3 K=1,25                                                       MAT10050
      KK=K                                                              MAT10060
      ALOW=AHIGH                                                        MAT10070
      AHIGH=ALOW+AINT                                                   MAT10080
      IF(K.EQ.25) AHIGH=BIG                                             MAT10090
      IF((ARR(J).GE.ALOW).AND.(ARR(J).LE.AHIGH)) GO TO 77               MAT10100
3     CONTINUE                                                          MAT10110
77    ICAT(KK)=ICAT(KK)+1                                               MAT10120
2     CONTINUE                                                          MAT10130
      SUM=0.                                                            MAT10140
      DO 90 I=1,25                                                      MAT10150
90    SUM=SUM+ICAT(I)                                                   MAT10160
      SCAT=0.                                                           MAT10170
      DO 91 I=1,25                                                      MAT10180
      CAT(I)=FLOAT(ICAT(I))/SUM                                         MAT10190
      IF(CAT(I).GT.SCAT) SCAT=CAT(I)                                    MAT10200
      IF(SCAT.EQ.CAT(I)) ISCAT=I                                        MAT10210
91    CONTINUE                                                          MAT10220
      AINS=.02                                                          MAT10230
      DO 7 I=1,25                                                       MAT10240
      LM=25-I+1                                                         MAT10250
      NUM(LM)=0                                                         MAT10260
      AHIGH=0.                                                          MAT10270
      DO 8 J=1,50                                                       MAT10280
      JJ=J                                                              MAT10290
      ALOW=AHIGH                                                        MAT10300
      AHIGH=ALOW+AINS                                                   MAT10310
      IF((CAT(I).GT.ALOW).AND.(CAT(I).LE.AHIGH)) GO TO 20               MAT10320
8     CONTINUE                                                          MAT10330
      GO TO 7                                                           MAT10340
20    NUM(LM)=JJ*2-1                                                    MAT10350
7     CONTINUE                                                          MAT10360
      VAL=BIG                                                           MAT10370
      WRITE(6,101)                                                      MAT10380
      WRITE(6,822) VAL                                                  MAT10390
822   FORMAT(1X,F9.1,1X,1HI)                                            MAT10400
      DO 10 I=1,25                                                      MAT10410
       LM=25-I+1                                                        MAT10420
       JZ=CAT(LM)*10.                                                   MAT10430
       JY=CAT(LM)*100.-JZ*10                                            MAT10440
       JX=CAT(LM)*1000.-JZ*100.-JY*10.                                  MAT10450
      IF(JX.EQ.0) JX=10                                                 MAT10460
```

```
      IF(JY.EQ.0) JY=10                                                 MAT10470
       IF(JZ.EQ.0) JZ=10                                                MAT10480
      VAL=VAL-AINT                                                      MAT10490
101   FORMAT(1H1)                                                       MAT10500
      NUMB=NUM(I)                                                       MAT10510
      IF(NUMB.GT.110) NUMB=110                                          MAT10520
      IF(NUMB.EQ.0) GO TO 15                                            MAT10530
      NUMB=NUMB-1                                                       MAT10540
      WRITE(6,12) (DOT(K),K=1,NUMB),TERM                                MAT10550
12    FORMAT(1X,10X,1HI,120A1)                                          MAT10560
       WRITE(6,11)VAL,(DOT(K),K=1,NUMB),TERM,BLANK,DDT,ANN(JZ),ANN(JY)  MAT10570
     C,ANN(JX)                                                          MAT10571
11    FORMAT(1X,F9.1,1X,1HI,120A1)                                      MAT10590
      GO TO 10                                                          MAT10600
15    WRITE(6,80)                                                       MAT10610
      WRITE(6,81) VAL                                                   MAT10620
80    FORMAT(11X,1HI)                                                   MAT10630
81    FORMAT(1X,F9.1,1X,1HI)                                            MAT10640
10    CONTINUE                                                          MAT10650
      WRITE(6,13)                                                       MAT10660
13    FORMAT(12X,10(10H---------I))                                     MAT10670
      WRITE(6,14) (AVAL(I),I=1,10)                                      MAT10680
14    FORMAT(13X,10(7X,F3.1))                                           MAT10690
      AMEAN=SSUM/LIM                                                    MAT10700
      SSQ=0.                                                            MAT10710
      DO 800 I=1,LIM                                                    MAT10720
800   SSQ=SSQ+(AMEAN-ARR(I))**2                                         MAT10730
      SDUM=0.                                                           MAT10740
      DO 801 I=1,25                                                     MAT10750
      SDUM=SDUM+CAT(I)                                                  MAT10760
      IF(SDUM.LT.0.5) GO TO 801                                         MAT10770
      AMED=SMALL+(I-1)*AINT+AINT*(SDUM-CAT(I))/SDUM                     MAT10780
      GO TO 802                                                         MAT10790
801   CONTINUE                                                          MAT10800
802   CONTINUE                                                          MAT10810
      AMODE=SMALL+AINT*(ISCAT-1)+AINT/2.0                               MAT10820
      AVAR=SSQ/LIM                                                      MAT10830
      WRITE(6,810)                                                      MAT10840
810   FORMAT(1H )                                                       MAT10850
      WRITE(6,811) AMEAN,AVAR,AMED,AMODE                                MAT10860
811   FORMAT( 13H THE MEAN IS ,F10.2, 18H  THE VARIANCE IS ,F10.2,      MAT10870
     C      16H  THE MEDIAN IS ,F10.2, 14H  THE MODE IS ,F10.2)         MAT10871
      RETURN                                                            MAT10890
900   WRITE(6,101)                                                      MAT10900
      WRITE(6,902) BIG                                                  MAT10910
902   FORMAT(1X, 49H ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =,  MAT10920
     C  F12.4)                                                          MAT10921
      RETURN                                                            MAT10940
      END                                                               MAT10950


      BLOCK DATA                                                        MAT10960
       COMMON/IDD/ RUNID(20)                                            MAT10970
      COMMON/PARA/ NNODE,NARC                                           MAT10990
       COMMON/ARC2/ DUM(500),IDUM(500),DUMA(500)                        MAT11010
      COMMON/ARC1/ BDUM(500),JDUM(1500),CDUM(2500)                      MAT11030
       COMMON/NOD1/ KDUM(2100),ZDUM(1000),LDUM(500),YDUM(100)           MAT11050
      COMMON/TERNN/LLDUM(31),ABBB(2000),KKKL(1030)
      COMMON/TERNI/ KJDUM(11)                                           MAT11090
```

```
      DATA KJDUM/11*0/                                                  MAT11100
      DATA NNODE/0/,NARC/0/
      DATA DUM/500*0.0/
      DATA IDUM/500*0/
      DATA DUMA/500*0.0/
      DATA BDUM/500*0.0/
      DATA JDUM/1500*0/
      DATA CDUM/2500*0.0/
      DATA KDUM/2100*0/
      DATA ZDUM/1000*0.0/
      DATA LDUM/500*0/
      DATA YDUM/100*0.0/
      DATA LLDUM/31*0/
      DATA ABBB/2000*0.0/
      DATA KKKL/1030*0/
       DATA RUNID/20*4H    /                                            MAT10980
      END                                                               MAT11110


      SUBROUTINE DUMP1                                                  MAT11120
      COMMON/PARA/NNODE,NARC                                            MAT11130
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT11140
      WRITE(6,1)                                                        MAT11150
 1    FORMAT(1X, 21HCOMING THROUGH NODCHK)                              MAT11160
      WRITE(6,2) (ISTAT(K),K=1,NARC)                                    MAT11170
 2    FORMAT(10I5)                                                      MAT11180
      RETURN                                                            MAT11190
      END                                                               MAT11200


      SUBROUTINE DUMP2                                                  MAT11210
      COMMON/PARA/ NNODE,NARC                                           MAT11220
      COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)                      MAT11230
      WRITE(6,1)                                                        MAT11240
 1    FORMAT(1X, 21HCOMING THROUGH ARCCHK)                              MAT11250
      WRITE(6,2) (ISTAT(K),K=1,NARC)                                    MAT11260
 2    FORMAT(10I5)                                                      MAT11270
      RETURN                                                            MAT11280
      END                                                               MAT11290


      SUBROUTINE UNIFRM(IXT,A,B,C,X)
C
C     THIS SUBROUTINE WILL CALCULATE UNIFORM DISTRIBUTED
C     VARIABLES
C     PA ADDITION TO PROGRAM AT 8 DEC 70
C     CODE NUMBER IS 3
C
      IF( C .EQ. A ) GO TO 1
      IF( B .NE. 0.0 ) CALL TERM( 1001)
      CALL RANDU(IXT,IXY,VAL)
      IXT = IXY
      X = A + VAL * (C - A )
      RETURN
    1 X = A
      RETURN
      END
```

APPENDIX II

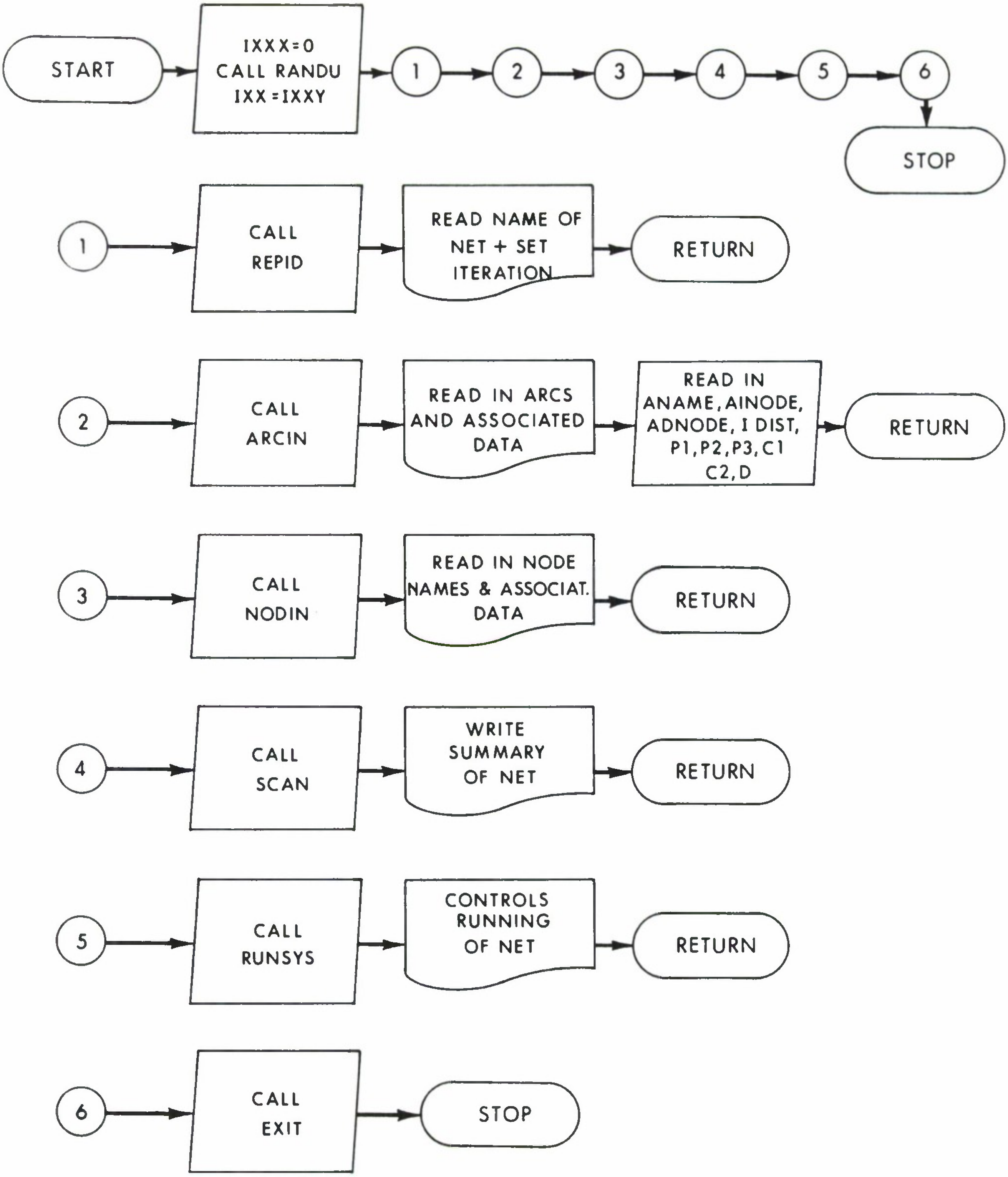
GENERAL FLOW CHART AND LISTING OF RISCA

Next page is blank.

This appendix includes a general processing flow chart for RISCA, but it does not include a description or flow chart of the individual subroutines. A detailed description and flow chart of the individual subroutines are provided in Appendix III. In addition to the flow chart, a complete program listing is provided. This version of RISCA is designed to run in a batch mode only, although the interested user could certainly adapt this program for a time sharing mode.

RISCA is written in FORTRAN IV. Even though FORTRAN IV is considered to be a standard language, adapting the program for any computer will probably require minor program modifications. These modifications generally result from peculiarities of the given system.

Unlike MATHNET, the RANDU subroutine in RISCA contains its' own uniform random number generating routine. Therefore, no modification in this area is required.



Flow Chart of "RISCA"

```
      COMMON/RANC/ IXX
C     PROGRAM RISCA
C     THIS IS THE MAIN ROUTINE                                          MAT00010
C                                                                       MAT00020
      IXXX=0
      CALL RANDU(IXXX,IXYY,RVAL)                                        MAT00110
      IXX=IXYY                                                          MAT00120
      CALL REPID
      CALL ARCIN
      CALL NODIN
      CALL SCAN
      CALL RUNSYS
      CALL EXIT
      STOP
      END                                                               MAT00590
      SUBROUTINE RANDU(I,J,RAN)
      IF(I.EQ.0)I=11111111
      J=I*25 $ J=J-(J/67108864)*67108864
      J=J*25 $ J=J-(J/67108864)*67108864
      J=J*5  $ J=J-(J/67108864)*67108864
      RAN=FLOAT(J)/67108864. $ RETURN $ END
      SUBROUTINE REPID
C     READS IN NAME OF NET AND SETS ITERATIONS
C
      COMMON/ITERA/ITER                                                 MAT05180
       COMMON/IDD/ RUNID(20)
       READ(5,2) RUNID
 2    FORMAT(25A4)
      ITER=500
       RETURN
       END
      SUBROUTINE ARCIN                                                  MAT07920
C     THIS ROUTINE WILL SERVE TO READ IN ARCS AND ASSOCIATED DATA       MAT07940
C                                                                       MAT07950
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON/PARA/NNODE,NARC                                            MAT07990
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT08010
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT08020
     2TIMEN(100)
      REAL NODE                                                         MAT08040
      INTEGER OARCI,OARC                                                MAT08050
      INTEGER ONODE                                                     MAT08080
      DATA RETC/4HRETU/                                                 MAT08060
C                                                                       MAT08070
 50   READ(5,1) ANAME,AINODE,AONODE,IDIST,P1,P2,P3,C1,C2,D              MAT08250
 1    FORMAT(3A4,I1,3F10.0,3F10.0)                                      MAT08260
      IF(ANAME.EQ.RETC) RETURN                                          MAT08270
      NARC=NARC+1                                                       MAT08280
      ARC(NARC)=ANAME                                                   MAT08290
      ITIMET(NARC)=IDIST                                                MAT08300
      TARG1(NARC)=P1                                                    MAT08310
      TARG2(NARC)=P2                                                    MAT08320
      TARG3(NARC)=P3                                                    MAT08330
```

```
      COSTC(NARC)=C1                                                    MAT08340
      COSTV(NARC)=C2                                                    MAT08350
      PROB(NARC)=D                                                      MAT0 13
      IF(NNODE.EQ.0) GO TO 10                                           MAT08370
      DO 9 I=1,NNODE                                                    MAT08380
      ISAVE=I                                                           MAT08390
C     ISAVE SIMPLY KEEPS VALUE OF I TO USE OUT OF DO LOOP               MAT08400
      IF(AINODE.EQ.NODE(I)) GO TO 11                                    MAT08410
 9    CONTINUE                                                          MAT08420
 10      NNODE=NNODE+1                                                  MAT08430
      NODE(NNODE)=AINODE                                                MAT08440
      INODE(NARC)=NNODE                                                 MAT08450
      OARC(NNODE,1)=NARC                                                MAT08460
      OARCI(NNODE)=1                                                    MAT08470
      GO TO 12                                                          MAT08480
 11   INODE(NARC)=ISAVE                                                 MAT08490
      OARCI(ISAVE)=OARCI(ISAVE)+1                                       MAT08500
      LM=OARCI(ISAVE)                                                   MAT08510
      OARC(ISAVE,LM)=NARC                                               MAT08520
 12   CONTINUE                                                          MAT08530
      DO 20 I=1,NNODE                                                   MAT08540
      ISAVE=I                                                           MAT08550
      IF(AONODE.EQ.NODE(I)) GO TO 30                                    MAT08560
 20   CONTINUE                                                          MAT08570
C     WE HAVE A NEW NODE                                                MAT08580
      NNODE=NNODE+1                                                     MAT08590
      NODE(NNODE)=AONODE                                                MAT08600
      ONODE(NARC)=NNODE                                                 MAT08610
      IARC(NNODE,1)=NARC                                                MAT08620
      IARCI(NNODE)=1                                                    MAT08630
      GO TO 40                                                          MAT08640
 30   ONODE(NARC)=ISAVE                                                 MAT08650
      IARCI(ISAVE)=IARCI(ISAVE)+1                                       MAT08660
      LM=IARCI(ISAVE)                                                   MAT08670
      IARC(ISAVE,LM)=NARC                                               MAT08680
 40   CONTINUE                                                          MAT08690
      GO TO 50
      END                                                               MAT08710
      SUBROUTINE NODIN                                                  MAT00600
C     THIS ROUTINE WILL READ IN NODE NAMES AND DATA                     MAT00610
C                                                                       MAT00620
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT00640
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT00650
     2TIMEN(100)
      REAL NODE                                                         MAT00670
      DIMENSION DDUM(10)                                                MAT00680
      COMMON/PARA/ NNODE,NARC                                           MAT00690
       COMMON/TERNN/ NODN(30),NODI,TIMEZ(500),COSTZ(500),NODEZ(500),
     1NCOUNT(30)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      DIMENSION ZNAM(10),ZPROB(10),IZNAM(10),YI(10),YO(10)              MAT00750
      INTEGER OARC,OARCI                                                MAT00770
      COMMON/TERNI/INODI(10),INODT                                      MAT00790
      INTEGER OTYPE                                                     MAT00800
      DATA CZZ/4HZZZZ/                                                  MAT00760
      DATA RETC/4HRETU/                                                 MAT00780
```

```
C
 20     READ(5,40) ADUM,IDUM,IODUM
 40     FORMAT(A4,I1,I1)
        IF(ADUM.EQ.RETC) RETURN
        DO 100 I=1,NNODE
        NDUM=I
        IF(NODE(I).EQ.ADUM) GO TO 101
 100    CONTINUE
        NNODE=NNODE+1
        NDUM=NNODE
 101    ITYPE(NDUM)=IDUM
        OTYPE(NDUM)=IODUM
        IF(ITYPE(NDUM).EQ.4) GO TO 90
C       ITYPE=4 MEANS INITIAL NODE
        IF(ITYPE(NDUM).EQ.6) GO TO 400
C       ITYPE=6 MEANS WE HAVE A 1/1 NODE WITH A NEGATIVE INPUT
        IF(ITYPE(NDUM).EQ.5) GO TO 60
C       ITYPE= 5 MEANS A 1/1 NODE WHICH MEANS WE MUST READ IN ORDERINGS
 30     IF(OTYPE(NDUM).EQ.2) GO TO 70
C       OTYPE=2 MEANS PROBABILISTIC FIRINGS..WE MUST READ IN PROBABILITYS
        IF(OTYPE(NDUM).EQ.4) GO TO 80
        IF(OTYPE(NDUM).EQ.7) GO TO 400
        GO TO 20
 60     CONTINUE
C       INSERT READS HERE FOR 1/1 NODES
        READ(5,303)  MM,(YI(I),YO(I),I=1,MM)
 303    FORMAT(I2,10(A4,A4))
 203    FORMAT(I2,10(A4,F6.3))
        DO 305 L=1,NARC
        DO 306 K=1,MM
        IF(ARC(L).EQ.YI(K)) IARC(NDUM,K)=L
        IF(ARC(L).EQ.YO(K)) OARC(NDUM,K)=L
 306    CONTINUE
 305    CONTINUE
        GO TO 20
 70     CONTINUE
C       INSERT READ FOR PROBABILITIES
 220    READ(5,203) NN,(ZNAM(I),ZPROB(I),I=1,NN)
        IF(NN.NE.OARCI(NDUM)) CALL TERM(1)
        DO 205 I=1,NN
        DO 206 J=1,NN
        JJ=J
        LM=OARC(NDUM,I)
        IF(ZNAM(J).EQ.ARC(LM)) GO TO 207
 206    CONTINUE
        CALL TERM(2)
 207    PROA(NDUM,I)=ZPROB(JJ)
 205    CONTINUE
        GO TO 20
 80     NODI=NODI+1
        NODN(NODI)=NDUM
        GO TO 20
 90         INODT=INODT+1
        INODI(INODT)=NDUM
        GO TO 30
  400   CONTINUE
 420    READ(5,303) MM,(YI(I),YO(I),I=1,MM)
        DO 405 L=1,NARC
```

```
      DO 406 K=1,MN                                                     MAI01920
      IF(ARC(L).EQ.YI(K)) IARC(NDUM,K)=L                                MAI01930
      IF(ARC(L).EQ.YO(K)) OARC(NDUM,K)=L                                MAI01940
      IF(YI(K).EQ.CZZ) IARC(NDUM,K)=100
 406  CONTINUE                                                          MAI01960
 405  CONTINUE                                                          MAI01970
      IARCI(NDUM)=IARCI(NDUM)+1                                         MAI01980
      GO TO 20                                                          MAI01990
      END                                                               MAI02180
      SUBROUTINE SCAN                                                   MAI07230
C     PRINTS A SUMMARY OF NET
C                                                                       MAI07930
      COMMON/ITERA/ITER
       COMMON/IDD/ RUNID(20)
      COMMON/RANC/ IXX                                                  MAI07250
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON/PARA/ NNODE,NARC                                           MAI07290
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAI07300
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAI07310
     2TIMEN(100)
       COMMON/TERNN/ NODN(30),NODI,TIMEZ(500),COSTZ(500),NODEZ(500),
     1NCOUNT(30)
      REAL NODE                                                         MAI07330
      INTEGER ONODE,OTYPE,OARCI                                         MAI07340
      DIMENSION TYPE(4),INRUL(7),OUTRUL(7),RUL(2)
      DATA TYPE/'NORM','TRI ','UNIF','CON '/
      DATA INRUL/'AND ','OR  ','    ','INIT','1-1 ','1-1B','PREF'/
      DATA OUTRUL/'ALL ','PROB','    ','TERM','1-1 ','1-1B','PREF'/
      DATA RUL/'NO  ','YES'/
C                                                                       MAI07350
      WRITE(6,4) RUNID
    4 FORMAT(1H1,25X,25A4)
      WRITE(6,7) ITER
    7 FORMAT(50X,I5,2X,'ITERATIONS')
      WRITE(6,50)
      WRITE(6,1)                                                        MAI07360
 1    FORMAT(1H0,1X,' ARC    INP NODE    OUT NODE    TIME DIST    ARG1',
     1'          ARG2          ARG3                   COST        P OF ',
     2'COMP')
      WRITE(6,50)
      DO 2 I=1,NARC                                                     MAI07390
      LM=INODE(I)                                                       MAI07400
      LN=ONODE(I)                                                       MAI07410
      LT=ITIMET(I)
      WRITE(6,3) ARC(I),NODE(LM),NODE(LN),TYPE(LT) ,TARG1(I)
     1,TARG2(I),TARG3(I),COSTC(I),COSTV(I),PROB(I)
 3    FORMAT(2X,A4,5X,A4,8X,A4,9X,A4 ,3(3X,F10.2),4X,F10.2,3H + ,F10.2,
     12H T,7X,F4.2)
 2    CONTINUE                                                          MAI07460
C                                                                       MAI07470
C     NOW WRITE OUT ARCS AND ASSOCIATED DATA                            MAI07480
      WRITE(6,50)
      WRITE(6,50)                                                       MAI07500
 50   FORMAT(1H )                                                       MAI07510
      WRITE(6,51)                                                       MAI07520
 51   FORMAT(1X,'   NODE        NO. OF INPUT ARCS   NO. OF OUTPUT ARCS',MAI07530
```

```
      1'     INPUT RULE     OUTPUT RULE ')
      WRITE(6,50)
      DO 60 I=1,NNODE                                                   MA)07550
      IR=ITYPE(I)
      IOR=OTYPE(I)
      WRITE(6,61) NODE(I),IARCI(I),OARCI(I),INROL(IR),OUTRUL(IOR)
   61 FORMAT(3X,A4,15X,I2,15X,I2,19X,A4,13X,A4)
   60 CONTINUE                                                          MAT07580
      RETURN                                                            MAT07590
      END                                                               MAT07600
      SUBROUTINE RUNSYS                                                 MAT05490
C     THIS ROUTINE CONTROLS THE RUNNING OF A NET                        MAT05500
C                                                                       MAT05510
       COMMON/IRRZ/IREPIT
      COMMON/KKIND/KIND                                                 MAT05520
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON/ITERA/ITER                                                 MAT05550
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/RUDER/ ITERMS                                              MAT05570
      COMMON/NINT/SMTIM,INSM                                            MAT05580
       COMMON/TERNN/ NODN(30),NODI,TIMEZ(500),COSTZ(500),NODEZ(500),
     1NCOUNT(30)
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT05610
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MA)05620
     2TIMEN(100)
      COMMON/PARA/NNODE,NARC                                            MAT05640
      COMMON/TERNI/ INODI(10),INODT                                     MAT05650
      INTEGER OARC,OARCI,OTYPE                                          MAT05660
C                                                                       MAT05670
C     SET NUMBER OF ITERATIONS SO FAR TO 0                              MAT05680
      IREPIT=0                                                          MAT05690
   20 SMTIM=999999.0                                                    MAT05710
C                                                                       MAT05730
C     ABOVE INITIALIZES SMALLEST TERMINAL FINISH TIME TO A LARGE NUMBER MAT05720
C     THE FOLLOWING CODE SETS THE INDICATER OF WHETHER OR NOT           MAT05740
C     A TERMINAL NODE HAS BEEN FILLED TO INDICATE NO.                   MAT05750
C                                                                       MAT05760
      ITERMS=0                                                          MAT05770
C                                                                       MAT05790
C     NOW FIRE INITIAL NODES                                            MAT05780
      DO 10 I=1,INODT                                                   MAT05800
      LM=INODI(I)                                                       MAT05810
      TIMEN(LM)=0.                                                      MAT05820
      IF(OTYPE(LM).EQ.1) CALL ALLFIR(LM)                                MAT05840
      IF(OTYPE(LM).EQ.2) CALL PROFIR(LM)                                MAT05850
   10 CONTINUE                                                          MAT05860
    1 CALL ARCCHK                                                       MAT05870
      CALL NODCHK                                                       MAT05880
      IF(KIND.EQ.0) GO TO 60                                            MAT05890
      IF(ITERMS.EQ.0) GO TO 1                                           MAT05900
      CALL ENDIT(KEY)                                                   MAT05930
      IF(KEY.EQ.0) GO TO 1                                              MAT05940
   61 CALL PTERM                                                        MAT05950
      IREPIT=IREPIT+1                                                   MAT05960
      IF(IREPIT.EQ.ITER) CALL SGRAPH                                    MAT05970
      IF(IREPIT.EQ.ITER) RETURN                                         MAT05980
      DO 40 I=1,NARC                                                    MAT05990
```

```
      TIME(I)=0.                                                        MAT06010
      ISTAT(I)=0                                                        MAT06020
 40   CONTINUE                                                          MAT06030
      DO 50 I=1,NNODF                                                   MAT06040
      TIMEN(I)=0.                                                       MAT06050
 50   CONTINUE                                                          MAT06070
      GO TO 20                                                          MAT06080
C     COME HERE IF NO NODES HAVE FIRED                                  MAT06090
 60   IF(ITERMS.EQ.0) CALL TERM(4)
C     DO ABOVE IF NO TERMINAL NODES HAVE BEEN FILLED                    MAT06110
      GO TO 61                                                          MAT06120
C     DO ABOVE IF THERE IS A TERMINAL NODE-END ITERATION                MAT06130
      END                                                               MAT06140
      SUBROUTINE ALLFIR(I)                                              MAT02800
C     FIRE ALL OUTPUT ARCS                                              MAT02810
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT02830
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT02840
     2TIMEN(100)
      REAL NODE                                                         MAT02860
      INTEGER OARCI,OARC                                                MAT02870
      II=OARCI(I)                                                       MAT02880
      DO 1 K=1,II                                                       MAT02890
      KK=OARC(I,K)                                                      MAT02900
      ISTAT(KK)=1                                                       MAT02910
 1    CONTINUE                                                          MAT02920
      RETURN                                                            MAT02950
      END                                                               MAT02960
      SUBROUTINE PROFIR(I)                                              MAT02550
C     FIRE NODE I USING STOCHASTIC CONSIDERATIONS                       MAT02560
C                                                                       MAT07750
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT02580
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT02590
     2TIMEN(100)
      REAL NODE                                                         MAT02610
      COMMON/RANC/ IXX                                                  MAT02620
      INTEGER OARC,OARCI,OTYPE                                          MAT02630
C                                                                       MAT02640
      IXXX=IXX                                                          MAT02650
      CALL RANDU(IXXX,IXY,RVAL)                                         MAT02660
      IXX=IXY                                                           MAT02670
      AHIGH=0.                                                          MAT02680
      II=OARCI(I)                                                       MAT02690
      DO 1 K=1,II                                                       MAT02700
      KK=OARC(I,K)                                                      MAT02710
      ALOW=AHIGH                                                        MAT02720
      AHIGH=ALOW+PPOA(I,K)                                              MAT02730
      IF((RVAL.GE.ALOW).AND.(RVAL.LE.AHIGH)) GO TO 2                    MAT02740
 1    CONTINUE                                                          MAT02750
      CALL TERM(3)
 2    ISTAT(KK)=1                                                       MAT02770
      RETURN                                                            MAT02780
      END                                                               MAT02790
      SUBROUTINE ARCCHK                                                 MAT02190
C     IF INITIATED&CHECK PROBABILITY OF COMPLETION                      MAT02200
C     IF COMPLETED&CALCULATE TIME AND COST                              MAT02210
C                                                                       MAT02220
```

```
      COMMON/RANC/ IXX                                                  MAT02230
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/ARC1/APC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON/PARA/ NNODE,NARC                                           MAT02270
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT02280
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT02290
     2TIMEN(100)
      REAL NODE                                                         MAT02310
C
      DO 1 I=1,NARC                                                     MAT02320
      IXXX=IXX                                                          MAT02330
      A=TARG1(I)                                                        MAT02340
      B=TARG2(I)                                                        MAT02350
      C=TARG3(I)                                                        MAT02360
      IF(ITIMET(I).EQ.1) CALL GAUSS(IXXX,B,A,TVAL)                      MAT02370
      IF(ITIMET(I).EQ.2)CALL TRIANG(IXXX,A,B,C,TVAL)                    MAT02380
      IF(ITIMET(I).EQ.3) CALL UNIF(IXXX,A,B,TVAL)                       SEP 70
      IF(ITIMET(I).EQ.4) TVAL=A
      IXX=IXXX                                                          MAT02390
      LM=INODE(I)                                                       MAT02400
      TIME(I)=TVAL+TIMEN(LM)                                            MAT02410
      IF(ISTAT(I).NE.1) GO TO 1                                         MAT02430
      IF(PROB(I).EQ.1.)GO TO 2                                          MAT02440
      IXXX=IXX                                                          MAT02450
      CALL RANDU(IXXX,IXYY,RVAL)                                        MAT02460
      IXX=IXYY                                                          MAT02470
      IF(RVAL.LE.PROB(I)) GO TO 2                                       MAT02480
      ISTAT(I)=3                                                        MAT02490
      GO TO 1                                                           MAT02500
  2   ISTAT(I)=2                                                        MAT02510
  1   CONTINUE                                                          MAT02520
      RETURN                                                            MAT02530
      END                                                               MAT02540
      SUBROUTINE GAUSS(IX,S,AM,V)                                       MAT05400
C     CALCULATES NORMAL DISTRIBUTED VARIABLES
C
      A=0.0                                                             MAT05410
      DO 50 I=1,12                                                      MAT05420
      CALL RANDU(IX,IY,Y)                                               MAT05430
      IX=IY                                                             MAT05440
 50   A=A+Y                                                             MAT05450
      V=(A-6.0)*S+AM                                                    MAT05460
      RETURN                                                            MAT05470
      END                                                               MAT05480
      SUBROUTINE TRIANG(IXT,A,B,C,X)                                    MAT07110
C     THIS ROUTINE WILL CALCULATE RANDUM TRIANGULARLY DISTRIBUTED       MAT07190
C     VARIABLES                                                         MAT07200
C                                                                       MAT07180
      IF((C-A).EQ.0.) CALL TERM(9)
      AM=(B-A)/(C-A)                                                    MAT07120
      CALL RANDU(IXT,IXY,VAL)                                           MAT07130
      IXT=IXY                                                           MAT07140
      IF(VAL.LE.AM) XI=SQRT(AM*VAL)                                     MAT07150
      IF(VAL.GT.AM) XI=1.-SQRT(1.-AM-VAL+AM*VAL)                        MAT07160
      X=A+XI*(C-A)                                                      MAT07170
      RETURN                                                            MAT07210
      END                                                               MAT07220
```

```
      SUBROUTINE UNIF(IX,OPT,PES,VAL)                                   SEP 70
C     CALCULATES UNIFORMLY DISTRIBUTED VARIABLES
C
      CALL RANDU(IX,IY,Y)                                               SEP 70
      IX=IY                                                             SEP 70
      VAL=OPT+Y*(PES-OPT)                                               SEP 70
      RETURN                                                            SEP 70
      END                                                               SEP 70
      SUBROUTINE NODCHK                                                 MAT02970
C     SEE WHAT NODES ARE READY TO FIRE,FIRE THOSE THAT                  MAT02980
C     ARE READY                                                         MAT02990
C                                                                       MAT03000
      COMMON/PARA/NNODE,NARC                                            MAT03010
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT03020
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT03030
     2TIMEN(100)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      INTEGER OTYPE,OARCI,OARC                                          MAT03060
      COMMON /KKIND/KIND                                                MAT03070
      REAL NODE                                                         MAT03080
C                                                                       MAT05700
      KIND=0                                                            MAT03090
      DO 1 I=1,NNODE                                                    MAT03100
      IL=I                                                              MAT03110
      J=0                                                               MAT03120
      IF(ITYPE(I).EQ.1) CALL ANDTST(IL,J)                               MAT03130
      IF(ITYPE(I).EQ.2) CALL ORTST(IL,J)                                MAT03140
      IF(ITYPE(I).EQ.5) GO TO 30                                        MAT03160
      IF(ITYPE(I).EQ.6) GO TO 31                                        MAT03170
       IF(ITYPE(I).EQ.7) GO TO 32
      IF(J.EQ.0) GO TO 1                                                MAT03180
      KIND=KIND+1                                                       MAT03190
C                                                                       MAT03200
C     NOW ZERO OUT INPUT ARCS                                           MAT03210
      IRK=IARCI(I)                                                      MAT03230
      DO 22 IJ=1,IRK                                                    MAT03240
      LM=IARC(I,IJ)                                                     MAT03250
       IF(ISTAT(LM).EQ.0) GO TO 22
       IF(ISTAT(LM).EQ.2) ISTAT(LM)=4
       IF(ISTAT(LM).EQ.3) ISTAT(LM)=4
       IF(ISTAT(LM).EQ.1) ISTAT(LM)=4
 22   CONTINUE                                                          MAT03270
      IF(OTYPE(I).EQ.1) CALL ALLFIR(I)                                  MAT03280
      IF(OTYPE(I).EQ.2) CALL PROFIR(I)                                  MAT03290
      IF(OTYPE(I).EQ.4) CALL ITALL(I)                                   MAT03300
 30   IF(OTYPE(I).EQ.5) CALL ONEONE(IL,J)
 31   IF(OTYPE(I).EQ.6) CALL ONEBAR(IL,J)
 32   IF(OTYPE(I).EQ.7) CALL PREFER(IL,J)
      IF(OTYPE(I).EQ.5) KIND=KIND+J
      IF(OTYPE(I).EQ.6) KIND=KIND+J
      IF(OTYPE(I).EQ.7) KIND=KIND+J
 1    CONTINUE                                                          MAT03330
      RETURN                                                            MAT03340
      END                                                               MAT03350
      SUBROUTINE ANDTST(I,J)                                            MAT04950
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT04970
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT04980
     2TIMEN(100)
```

```
      REAL NODE                                                         MAT05000
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
C                                                                       MAT07490
      II=IARCI(I)                                                       MAT05030
      TIM=0.                                                            MAT05040
      DO 1 K=1,II                                                       MAT05050
      KK=IARC(I,K)                                                      MAT05060
      IF(ISTAT(KK).NE.2) GO TO 2                                        MAT05070
      IF(TIME(KK).GT.TIM) TIM=TIME(KK)                                  MAT05080
 1    CONTINUE                                                          MAT05100
      TIMEN(I)=TIM                                                      MAT05110
      J=1                                                               MAT05120
      RETURN                                                            MAT05130
 2    J=0                                                               MAT05140
      RETURN                                                            MAT05150
      END                                                               MAT05160
      SUBROUTINE ORTST(I,J)                                             MAT04660
C     THIS ROUTINE TEST 'OR' NODES                                      MAT04670
C                                                                       MAT06660
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT04680
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT04690
     2TIMEN(100)
      REAL NODE                                                         MAT04710
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
C                                                                       MAT06670
      II=IARCI(I)                                                       MAT04750
      TIM=1000000.                                                      MAT04760
      J=0                                                               MAT04770
      DO 1 K=1,II                                                       MAT04780
      KK=IARC(I,K)                                                      MAT04790
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAT04800
C     THIS ROUTINE TESTS'AND' NODES                                     MAT04960
C                                                                       MAT03220
      J=1                                                               MAT04810
      IF(TIME(KK).LT.TIM) TIM=TIME(KK)                                  MAT04820
 1    CONTINUE                                                          MAT04830
      IF(J.EQ.0) RETURN                                                 MAT04840
      TIMEN(I)=TIM                                                      MAT04850
      RETURN                                                            MAT04930
      END                                                               MAT04940
       SUBROUTINE ONEONE(ILK,J)
C     TESTS ONE-ONE NODES
C
      COMMON/PARA/NNODE,NARC                                            MAT03370
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT03380
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT03390
     2TIMEN(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      INTEGER OTYPE,OARCI,OARC                                          MAT03440
      REAL NODE                                                         MAT03450
C
      J=0                                                               MAT03460
      II=IARCI(ILK)                                                     MAT03470
      TIM=1000000.                                                      MAT03480
```

```
      DO 1 K=1,II                                                       MAI03490
      KK=IARC(ILK,K)                                                    MAI03500
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAI03510
      IF(TIME(KK).GE.TIM) GO TO 1                                       MAI03520
      TIM=TIME(KK)                                                      MAI03530
      J=K                                                               MAI03540
 1    CONTINUE                                                          MAI03550
      IF(J.EQ.0) RETURN                                                 MAI03560
C     ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC                         MAI03570
      DO 2 K=1,II                                                       MAI03580
      LM=IARC(ILK,K)                                                    MAI03590
       IF(ISTAT(LM).EQ.0) GO TO 2
       ISTAT(LM)=4
 2    CONTINUE                                                          MAI03610
      LM=OARC(ILK,J)                                                    MAI03620
      ISTAT(LM)=1                                                       MAI03630
      TIMEN(ILK)=TIM                                                    MAI03640
      RETURN                                                            MAI03720
      END                                                               MAI03730
       SUBROUTINE ONEBAR(ILK,J)
C     TESTS ONE-ONE BAR NODES
C
      COMMON/PAPA/NNODE,NARC                                            MAI03960
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAI03970
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAI03980
     2TIMEN(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      INTEGER OTYPE,OARCI,OARC                                          MAI04030
      REAL NODE                                                         MAI04040
C
      J=0                                                               MAI04050
      II=IARCI(ILK)                                                     MAI04060
      TIM=1000000.                                                      MAI04070
      DO 1 K=1,II                                                       MAI04080
      KK=IARC(ILK,K)                                                    MAI04090
      IF(KK.EQ.100) IBAR=K
      IF(KK.EQ.100) GO TO 1
      IF(ISTAT(KK).NE.2) GO TO 1                                        MAI04120
      IF(TIME(KK).GE.TIM) GO TO 1                                       MAI04130
      TIM=TIME(KK)                                                      MAI04140
      J=K                                                               MAI04150
 1    CONTINUE                                                          MAI04160
      IF(J.EQ.0) GO TO 5                                                MAI04170
C     ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC                         MAI04180
      DO 2 K=1,II                                                       MAI04190
      LM=IARC(ILK,K)                                                    MAI04200
       IF(ISTAT(LM).EQ.0) GO TO 2
       ISTAT(LM)=4
 2    CONTINUE                                                          MAI04220
      LM=OARC(ILK,J)                                                    MAI04230
      ISTAT(LM)=1                                                       MAI04240
      TIMEN(ILK)=TIM                                                    MAI04250
      RETURN                                                            MAI04340
 5     TIM=0.
      DO 6 K=1,II                                                       MAI04370
      KK=IARC(ILK,K)                                                    MAI04380
```

```
      IF(KK.EQ.100) GO TO 6
      IF(ISTAT(KK).NE.3) GO TO 7                                        MAT04400
       IF(TIME(KK).LE.TIM) GO TO 6
      TIM=TIME(KK)                                                      MAT04420
 6    CONTINUE                                                          MAT04430
      LM=OARC(ILK,IBAR)                                                 MAT04440
      ISTAT(LM)=1                                                       MAT04450
       J=1
      DO 12 K=1,II                                                      MAT04460
      LM=IARC(ILK,K)                                                    MAT04470
       ISTAT(LM)=4
 12   CONTINUE                                                          MAT04490
      TIMEN(ILK)=TIM                                                    MAT04500
 7    CONTINUE                                                          MAT04630
      RETURN                                                            MAT04640
      END                                                               MAT04650
       SUBROUTINE PREFER(ILK,J)
C     TESTS PREFERENCE NODES
C
      COMMON/PARA/NNODE,NARC
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
     2TIMEN(100)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      INTEGER OTYPE,OARCI,OARC
      REAL NODE
       J=0
       II=IARCI(ILK)
       DO 1 K=I,II
       KK=IARC(ILK,K)
       IF(KK.EQ.100) GO TO 1
       IF(ISTAT(KK).EQ.0) GO TO 30
       IF(ISTAT(KK).EQ.1) GO TO 30
       IF(ISTAT(KK).EQ.4) GO TO 30
 1     CONTINUE
       J=1
C     IF WE GET HERE THE NODE WILL BE FIRED
C      FIRE FIRST ARC PAIR WITH 2 STATUS, IF THERE IS ONE
       TIM=0.
      III=II-I
       DO 2 K=1,III
       KK=IARC(ILK,K)
       KKK=K
       IF(TIME(KK).GT.TIM) TIM=TIME(KK)
       IF(ISTAT(KK).EQ.2) GO TO 4
 2     CONTINUE
C     IF WE GET HERE FIRE BAR ARC
       LM=OARC(ILK,II)
       GO TO 5
C      FIRE THE KKK ARC
 4     LM=OARC(ILK,KKK)
 5     ISTAT(LM)=1
       DO 40 K=1,II
       LM=IARC(ILK,K)
       ISTAT(LM)=4
 40    CONTINUE
```

```
      TIMEN(ILK)=TIN
 30   CONTINUE
      RETURN
      END
      SUBROUTINE ITALL(II)                                              MAT07740
C     THIS ROUTINE WILL HANDLE A TERMINAL NODE BEING FILLED             MAT07760
C     IT WILL SEE IF THE TIME IS SMALLER THAN ANY OTHER TERMINAL        MAT07770
C     NODE TIME AND IF SO SWAP TIME AND COST INDICATORS                 MAT07780
C                                                                       MAT07790
      COMMON/MINT/SHTIM,INSM                                            MAT07800
      COMMON/RUNER/ ITERMS                                              MAT07810
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT07820
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT07830
     2TIMEN(100)
      ITERMS=1                                                          MAT07850
C     ABOVE INDICATES A TERMINAL NODE HAS BEEN FILLED                   MAT07860
      IF(TIMEN(II).GE.SMTIM) RETURN                                     MAT07870
      SMTIM=TIMEN(II)                                                   MAT07880
      INSM=II                                                           MAT07890
      RETURN                                                            MAT07900
      END                                                               MAT07910
      SUBROUTINE ENDIT(KEY)                                             MAT06370
C                                                                       MAT06380
C     THIS ROUTINE CHECKS TO SEE IF THERE ARE COMPLETED ARCS WITH TIMES MAT06390
C     SMALLER THAN THE SMALLEST TERMINAL NODE                           MAT06400
C     IF SO SET KEY=0, IF NOT SET KEY=1 AND TERMINATE THIS RUN          MAT06410
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/MINT/ SMTIM,INSM                                           MAT06430
      COMMON/PARA/NNODE,NARC                                            MAT06440
      DO 1 I=1,NARC                                                     MAT06450
      IF(ISTAT(I).EQ.0) GO TO 1
      IF(ISTAT(I).EQ.3) GO TO 1
      IF(ISTAT(I).EQ.4) GO TO 1
      IF(TIME(I).LT.SMTIM) GO TO 2                                      MAT06470
 1    CONTINUE                                                          MAT06480
      KEY=1                                                             MAT06490
      RETURN                                                            MAT06500
 2    KEY=0                                                             MAT06510
      RETURN                                                            MAT06520
      END                                                               MAT06530
      SUBROUTINE PTERM                                                  MAT06150
C     DETERMINES COST OF ITERATION
C
       COMMON/PARA/ NNODE,NARC
       COMMON/IRRZ/IREPIT
      COMMON/MINT/ SMTIM,INSM                                           MAT06160
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),    MAT06170
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAT06180
     2TIMEN(100)
       COMMON/TERNN/ NODN(30),NODI,TIMEZ(500),COSTZ(500),NODEZ(500),
     1NCOUNT(30)
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
       COMMON /IDD/ RUNID(20)
      COMMON/TERNI/ INODI(10),INODT                                     MAT06220
C                                                                       MAT06230
      DO 1 I=1,NODI                                                     MAT06240
```

```
      JJ=I                                                              MAI06250
      IF(NODN(I).EQ.INSM) GO TO 2                                       MAI06260
 1    CONTINUE                                                          MAI06270
      CALL TERM(69)                                                     MAI06280
 2    NCOUNT(JJ)=NCOUNT(JJ)+1                                           MAI06290
      LM=NCOUNT(JJ)                                                     MAI06300
       IRR=IREPIT+1
       TIMEZ(IRR)=SMTIM
  6    NODEZ(IRR)=INSM
       COSTZ(IRR)=0.
       DO 3 I=1,NARC
       LM=INODE(I)
       IF(ISTAT(I).EQ.0) GO TO 3
       IF(TIME(I).GT.SMTIM) GO TO 4
       COSTZ(IRR)=COSTZ(IRR)+COSTC(I)+COSTV(I)*(TIME(I)-TIMEN(LM))
       GO TO 3
 4     IF(TIMEN(LM).GT.SMTIM) GO TO 3
       COSTZ(IRR)=COSTZ(IRR)+COSTC(I)+COSTV(I)*(SMTIM-TIMEN(LM))
 3     CONTINUE
       RETURN                                                           MAI06350
       END                                                              MAI06360
       SUBROUTINE SGRAPH                                                MAI06540
        DIMENSION DOT(120)
        COMMON/TERNN/ NODN(30),NODI,TIMEZ(500),COSTZ(500),NODEZ(500),
     1NCOUNT(30)
       REAL NODE                                                        MAI06570
       COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),   MAI06580
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),           MAI06590
     2TIMEN(100)
       COMMON/ITERA/ ITER                                               MAI06620
        DIMENSION ARRR(500)
        DIMENSION ANN(10)
        DIMENSION NBLIP(10)
        DIMENSION AVAL(10)
        COMMON/IDD/ RUNID(20)
        DATA TT/1HI/
        DATA DOT/120*1H=/
        DATA ANN/4H1111,4H2222,4H3333,4H4444,4H5555,4H6666,4H7777,4H8888,
     14H9999,4H0000/
        DATA DDT/4H..../
        DATA AVAL/.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0/
C                                                                       MAI06640
C      FIRST GENERATE INDIVIDUAL GRAPHS BY NODE                         MAI06650
        DO 1 J=1,NODI
        LM=NODN(J)
        LL=NCOUNT(J)
        DO 3 IZ=1,2
        LLK=0
        DO 2 K=1,ITER
        IF(NODEZ(K).NE.LM) GO TO 2
        LLK=LLK+1
        IF(IZ.EQ.1) ARRR(LLK)=TIMEZ(K)
        IF(IZ.EQ.2) ARRR(LLK)=COSTZ(K)
 2      CONTINUE
        IF(LLK.NE.LL) WRITE(6,109) LLK,LL
 109    FORMAT(1X,'VALUES OF LLK AND LL ARE ',2I5)
        IF(LLK.NE.LL) CALL TERM(6)
        CALL GRAPH(ARRR,LL)
```

```
      WRITE(6,60)
      IF(IZ.EQ.1) WRITE(6,61) NODE(LM)
      IF(IZ.EQ.2) WRITE(6,62) NODE(LM)
      WRITE(6,60)
      WRITE(6,25) RUNID
60    FORMAT(1H )
61    FORMAT(20X,'GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ',A4)
62    FORMAT(20X,'GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ',A4)
3      CONTINUE
1     CONTINUE                                                          MAT06870
      CALL GRAPH(TIMEZ,ITER)
      WRITE(6,10)
10    FORMAT(1H )                                                       MAT06790
      WRITE(6,10)
      WRITE(6,65)
65    FORMAT(20X,'GRAPH OF COMPLETION TIMES FOR ALL NODES')
      WRITE(6,10)
      WRITE(6,25) RUNID
      CALL GRAPH(COSTZ,ITER)
      WRITE(6,10)
      WRITE(6,10)
      WRITE(6,66)
66    FORMAT(20X,'GRAPH OF COMPLETION COSTS FOR ALL NODES')
      WRITE(6,10)
      WRITE(6,25) RUNID
25     FORMAT(20X,25A4)
      WRITE(6,111)
111   FORMAT(1H1)
      DO 200 JJ=1,NODI
      NCC=NCOUNT(JJ)
200   NBLIP(JJ)=FLOAT(NCC)*100./FLOAT(ITER)
      DO 201 JJ=1,NODI
       JZ=NBLIP(JJ)/10
      JY=NBLIP(JJ)-JZ*10
      IF(JZ.EQ.0) JZ=10
      IF(JY.EQ.0) JY=10
      KK=NODN(JJ)
      LL=NBLIP(JJ)
      IF(LL.EQ.0) GO TO 201
      WRITE(6,202) (DOT(K),K=1,LL),TT
      WRITE(6,203) NODE(KK),(DOT(K),K=1,LL),TT,DDT,ANN(JZ),ANN(JY)
      WRITE(6,204)
      WRITE(6,204)
202   FORMAT(11X,1HI,120A1)
203   FORMAT(6X,A4,1X,1HI,120A1)
204   FORMAT(11X,1HI)
201   CONTINUE
      WRITE(6,13)
13    FORMAT(12X,10(10H---------I))
      WRITE(6,14) (AVAL(I),I=1,10)
14    FORMAT(13X,10(7X,F3.1))
      WRITE(6,10)
      WRITE(6,10)
      WRITE(6,15)
15    FORMAT(20X,'GRAPH OF NODE PROBABILITIES')
       WRITE(6,10)
      WRITE(6,25) RUNID
      WRITE(6,111)
```

```
      RETURN                                                            MAT07090
      END                                                               MAT07100
      SUBROUTINE GRAPH(ARR,LIM)
      DIMENSION ANN(10)                                                 MAT09070
      DIMENSION ARR(LIM)                                                MAT09100
      COMMON/NOD1/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
     1IARCI(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
     2TIMEN(100)
       COMMON/IDD/ RUNID(20)
      DIMENSION CAT(50)                                                 MAT09110
      DIMENSION ICAT(50),NUM(60),DOT(120),AVAL(10)                      MAT09130
       DATA DOT/4H..../
       DATA ANN/4H1111,4H2222,4H3333,4H4444,4H5555,4H6666,4H7777,4H8888,
     14H9999,4H0000/
      DATA BLANK/1H /                                                   MAT09090
      DATA TERM/1HI/                                                    MAT09120
      DATA AVAL/.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0/                         MAT09140
      DATA DOT/120*1H=/                                                 MAT09150
      BIG=0.                                                            MAT09160
       SMALL=10000000.
      AVE=0.0
      SD=0.0
      IF(LIM.EQ.0) GO TO 900
      DO 1 I=1,LIM                                                      MAT09180
      AVE = AVE+ARR(I)
      IF(ARR(I).GT.BIG) BIG=ARR(I)                                      MAT09190
      IF(ARR(I).LT.SMALL) SMALL=ARR(I)                                  MAT09200
1     CONTINUE                                                          MAT09210
      RANGE=BIG-SMALL                                                   MAT09220
      IF(RANGE.EQ.0.) GO TO 900
      AVE = AVE/FLOAT(LIM)
      AINT=RANGE/25.                                                    MAT09240
      DO 4 K=1,50                                                       MAT09250
4     ICAT(K)=0                                                         MAT09260
      DO 2 J=1,LIM                                                      MAT09270
      SD=SD+((AVE-ARR(J))**2)
      AHIGH=SMALL                                                       MAT09280
      DO 3 K=1,26                                                       MAT09290
      ALOW=AHIGH                                                        MAT09300
      AHIGH=ALOW+AINT                                                   MAT09310
      IF((ARR(J).GE.ALOW).AND.(ARR(J).LE.AHIGH)) ICAT(K)=ICAT(K)+1      MAT09320
3     CONTINUE                                                          MAT09330
2     CONTINUE                                                          MAT09340
      IF(LIM.GT.1) VAR=SD/(FLOAT(LIM)-1.0)
      IF(LIM.LE.1) VAR=0
      SD=SQRT(VAR)
      SUM=0.                                                            MAT09350
      DO 90 I=1,26                                                      MAT09360
90    SUM=SUM+(FLOAT(ICAT(I)))
      INDEX=1
      DO 92 I=1,26
   92 CAT(I)=FLOAT(ICAT(I))/SUM
      GO TO 93
   94 INDEX=2
      DO 91 I=2,26
      CAT(I)=CAT(I)+CAT(I-1)
   91 CONTINUE
   93 AINS=.02
```

```
      DO 7 I=1,26                                                       MAT09410
      LM=26-I+1                                                         MAT09420
      NUM(LM)=0                                                         MAT09430
      AHIGH=0.                                                          MAT09440
      DO 8 J=1,51
      JJ=J                                                              MAT09460
      ALOW=AHIGH                                                        MAT09470
      AHIGH=ALOW+AINS                                                   MAT09480
      IF((CAT(I).GT.ALOW).AND.(CAT(I).LE.AHIGH)) GO TO 20               MAT09490
8     CONTINUE                                                          MAT09500
      GO TO 7                                                           MAT09510
20    NUM(LM)=JJ*2-1                                                    MAT09520
7     CONTINUE                                                          MAT09530
      VAL=BIG+AINT                                                      MAT09540
      WRITE(6,101)                                                      MAT09550
      DO 10 I=1,26                                                      MAT09560
       LM=26-I+1
       JZ=CAT(LM)*10.
       JY=CAT(LM)*100.-FLOAT(JZ)*10.
       JX=CAT(LM)*1000.-FLOAT(JZ)*100.-FLOAT(JY)*10.
      IF(JY.EQ.0) JY=10                                                 MAT09590
       IF(JZ.EQ.0) JZ=10
        IF(JX.EQ.0) JX=10
      VAL=VAL-AINT                                                      MAT09610
101   FORMAT(1H1)                                                       MAT09620
      NUMB=NUM(I)                                                       MAT09630
      IF(NUMB.GT.110) NUMB=110                                          MAT09640
      IF(NUMB.EQ.0) GO TO 15                                            MAT09650
      NUMB=NUMB-1                                                       MAT09660
      WRITE(6,12) (DOT(K),K=1,NUMB),TERM                                MAT09670
12    FORMAT(1X,12X,1HI,120A1)
       WRITE(6,11)VAL,(DOT(K),K=1,NUMB),TERM,BLANK,DDT,ANN(JZ),ANN(JY)
     1,ANN(JX)
11    FORMAT(1X,F11.3,1X,1HI,120A1)
      GO TO 10                                                          MAT09710
15    WRITE(6,80)                                                       MAT09720
      WRITE(6,81) VAL                                                   MAT09730
80    FORMAT(13X,1HI)
81    FORMAT(1X,F11.3,1X,1HI)
10    CONTINUE                                                          MAT09760
      WRITE(6,13)                                                       MAT09770
13    FORMAT(12X,10(10H---------I))                                     MAT09780
      WRITE(6,14) (AVAL(I),I=1,10)                                      MAT09790
14    FORMAT(13X,10(7X,F3.1))                                           MAT09800
      WRITE(6,70) AVE,VAR,SD
   70 FORMAT(1H0,16X,'MEAN =',F11.3,15X,'VARIANCE =',F11.3,15X,'STANDARD
     1 DEVIATION =',F11.3)
      IF(INDEX.EQ.1) GO TO 94
      RETURN                                                            MAT09810
900   WRITE(6,101)                                                      MAT09820
      IF(LIM.NE.0) WRITE(6,902) BIG
      IF(LIM.EQ.0) WRITE(6,903)
902   FORMAT(1X,' ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =',    MAT09840
     1F12.4)                                                            MAT09850
 903 FORMAT(1X,'THIS NODE WAS NEVER A TERMINAL NODE')
      RETURN                                                            MAT09860
      END                                                               MAT09870
      SUBROUTINE TERM(I)
```

```
C     THIS SUBROUTINE PRINTS ERROR MESSAGES
C
      COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
     1TARG2(100),TARG3(100),COSTC(100),COSTV(100)
      COMMON/IRRZ/IREPIT
      COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
      COMMON/PARA/NNODE,NARC
      GO TO (1,2,3,4,5,6,24,25,27),I
    1 WRITE(6,7)
    7 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - WRONG NUMBER OF
     1ARCS STATED***NODIN')
      CALL EXIT
    2 WRITE(6,8)
    8 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - WRONG ARC LISTED
     1***NODIN')
      CALL EXIT
    3 WRITE(6,9)
    9 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - NO ARC INITIATED
     1***PROFIR')
      CALL EXIT
    4 WRITE(6,10)
   10 FORMAT(1X,'***ERROR***NO INPUT RULES WERE SATISFIED***RUNSYS')
      WRITE(6,13) IREPIT
   13 FORMAT(1H0,17X,'ITERATION ',I3)
      WRITE(6,14)
   14 FORMAT(10X,'STATUS OF ALL ARCS FOLLOWS',/)
      DO 15 I=1,NARC
   15 WRITE(6,17) ARC(I),ISTAT(I)
   17 FORMAT(17X,A4,5X,I1)
      WRITE(6,18)
   18 FORMAT(10X,'WHERE0   0   NOT INITIATED',/19X,'1   INITIATED',/19X,
     1'2   COMPLETED SUCCESSFULLY',/19X,'3   FAILED',/19X,
     2'4   COST + TIME VALUES ALREADY CONSIDERED')
      CALL EXIT
    5 WRITE(6,11)
   11 FORMAT(1X,'***ERROR***COULD NOT DETERMINE TERMINAL NODE***PTERM')
      CALL EXIT
    6 WRITE(6,12)
   12 FORMAT(1X,'***ERROR***LLK AND LL MUST BE EQUAL***SGRAPH')
      CALL EXIT
   24 WRITE(6,23)
   23 FORMAT(1X,'***ERROR***NO TERMINAL NODES ARE PUNCHED TO PRINT')
      CALL EXIT
   25 WRITE(6,26)
   26 FORMAT(1X,'***ERROR***CAN NOT FIND TERMINAL NODES FOR SCAN')
      CALL EXIT
   27 WRITE(6,28)
   28 FORMAT(1X,'***ERROR***CHECK ALL ARC CARDS,'/11X,'AT LEAST ONE SHOW
     1S FIRST AND THIRD ARGUMENT EQUAL WHILE TIME DISTRIBUTION TYPE IS
     2TRIANGULAR'/11X,'CHANGE TO CONSTANT')
      CALL EXIT
      STOP                                                              MAT07720
      END
      BLOCK DATA                                                        MAT08720
        COMMON/IDD/ RUNID(20)
      COMMON/PARA/ NNODE,NARC                                           MAT08730
        COMMON/ARC2/ DUM(100),IDUM(100),DUMA(100)
        COMMON/TERNN/ LLDUM(31),ARDUM(1000),KLDUM(530)
```

```
      COMMON/ARC1/ BDUM(100),JDUM(300),CDUM(500)
       COMMON/LOD1/ KDUM(2100),ZDUM(1000),LDUM(500),YDUM(100)
      COMMON/TERM1/ KJDUM(11)                                           MAT08830
       DATA FUNID/20*4H    /
      DATA NNODE/0/,NARC/0/                                             MAT08740
       DATA DUM/100*0.0/,IDUM/100*0/,DUMA/100*0.0/
       DATA LLDUM/31*0/,ABDUM/1000*0.0/,KLDUM/530*0/
      DATA BDUM/100*0.0/,JDUM/300*0/,CDUM/500*0.0/
       DATA KDUM/2100*0/,ZDUM/1000*0.0/,LDUM/500*0/,YDUM/100*0.0/
      DATA KJDUM/11*0/                                                  MAT08840
      END                                                               MAT08850
*     LIST(STOP)
```

APPENDIX III

SUBROUTINES THAT MAKE UP MATHNET AND RISCA

Next page is blank.

This appendix includes a detailed flow chart for each of the 25 subroutines in RISCA**. Table I contains a listing of the 25 subroutines in the sequence of their appearance in the appendix. In addition to the name and sequence number, a brief description of the subroutine function is given in Table I.

TABLE I. SUBROUTINE FUNCTION

Name	Functions
1. Subroutine IDIN	Reads in run identifier card name of network.
2. Subroutine ARCIN	Reads in the Arc cards. Stores and orders Arc data into corresponding arrays.
3. Subroutine NODIN	Reads in the Node cards. Stores and orders Node data into corresponding arrays.
4. Subroutine SCAN	Summarizes the input data and prints out an input listing.
5. Subroutine RUNSYS	Runs the simulation and prints the results.
6. Subroutine REPSET	Sets the number of iteration.
7. Subroutine ALLFIR(I)	Fires* nodes using "AND" output arcs. Fires all output arcs from fix node.
8. Subroutine PROFIR(I)	Randomly selects an arc that exits from a probability node.

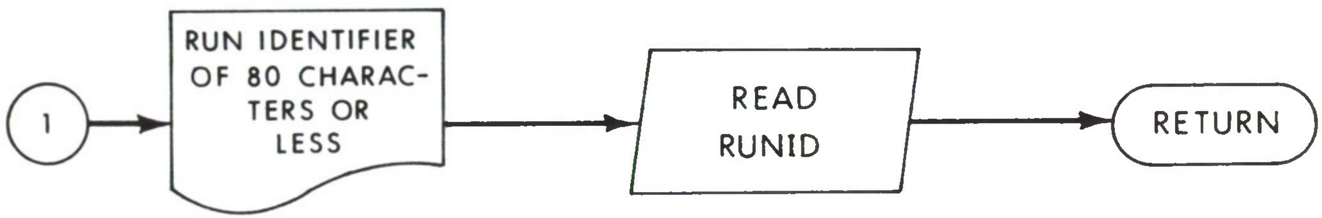
* Initiates
**Most of these subroutines are the same as those used in MATHNET.

TABLE I. SUBROUTINE FUNCTION (Continued)

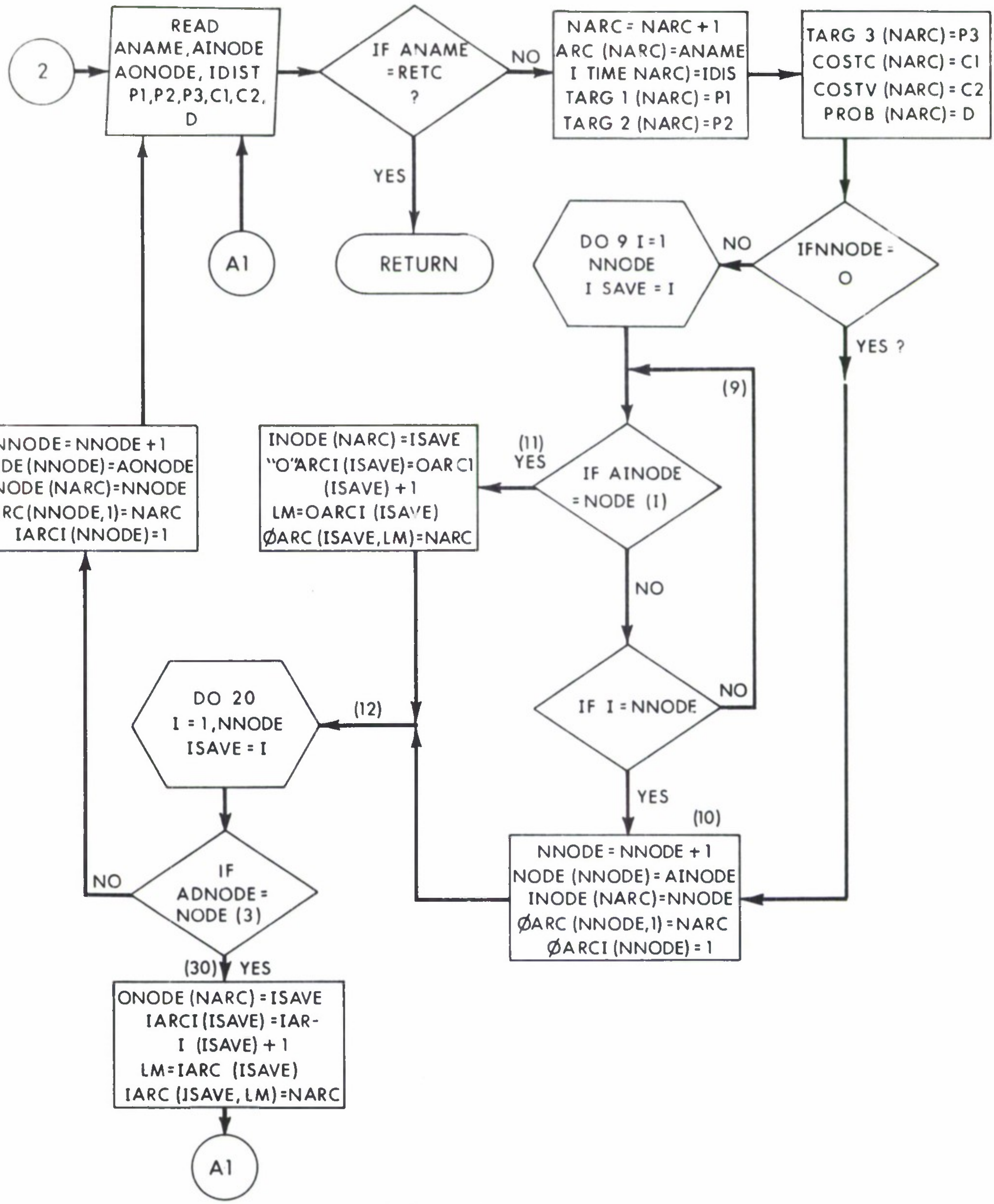
Name	Functions
9. Subroutine ARCCHK	Checks all of the arcs to see if they have been initiated. Checks the probability of successful completion of initiated arcs. Calculates the time for completion. 1 - normal distribution (sub GAUSS) 2 - triangular distribution (sub TRIANG) 3 - uniform distribution (sub UNIF)
10. Subroutine GAUSS	Calculates normal random variables.
11. Subroutine RANDU	Generates uniform random numbers.
12. Subroutine TRIANG	Calculates the triangularly distributed random variables.
13. Subroutine UNIF	Calculates the uniform random variables.
14. Subroutine NODCHK	Executes the output rules of those nodes whose input rules have just been satisfied by the arcs completed in subroutine ARCCHK. Determines what nodes are ready to fire. Fires those nodes that are ready.
15. Subroutine ANDTST	Tests "AND" nodes. Tests the time (t) taken by each arc and stores this information. Saves the time (t) of the longest arc to completion.
16. Subroutine ORTST	Tests "OR" nodes. Calculates cumulative time needed to satisfy input rule. Stores time (t) taken by each arc.

TABLE I. SUBROUTINE FUNCTION (Continued)

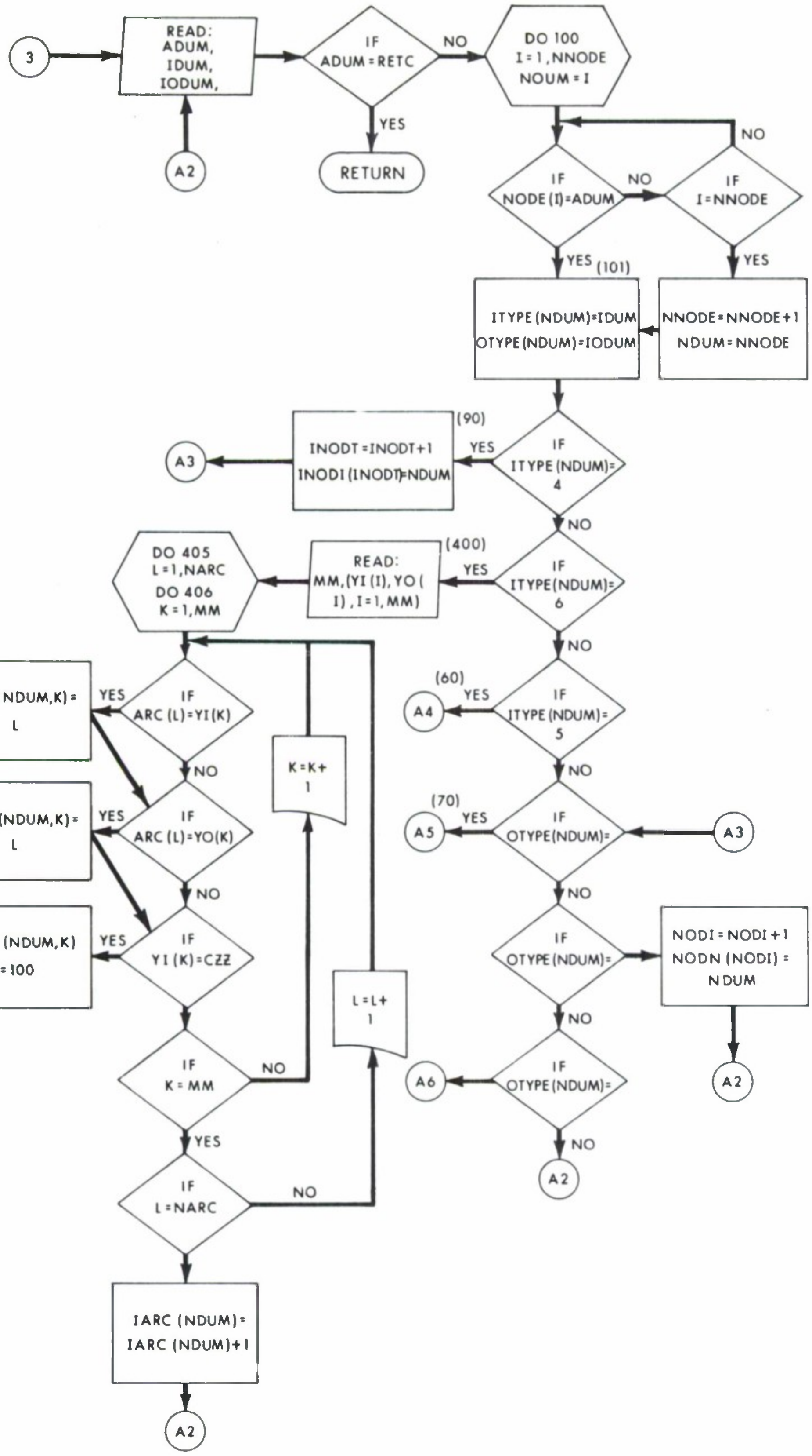
Name	Functions
17. Subroutine ONEONE	Tests and runs one-one nodes (1/1).
18. Subroutine ONEBAR	Tests and runs one-one bar nodes ($\overline{1/1}$).
19. Subroutine PREFER	Tests and runs preferred nodes.
20. Subroutine ITALL	This subroutine will handle all Terminal Node.
21. Subroutine ENDIT	Stores information on completed arcs.
22. Subroutine PTERM	Determines which terminal node was reached first in each iteration. Calculates how many times each terminal node was selected. Computes the total cost and completion time for each iteration.
23. Subroutine SGRAPH	Determines what is to be graphed and supplies titles for each graph. The actual graphing is done by subroutine GRAPH except for the final graph which lists the probabilities of reaching the various terminal nodes.
24. Subroutine GRAPH	Does the actual printing of the time and cost graphs.
25. Subroutine TERM	Prints all error messages in MATHNET and RISCA.



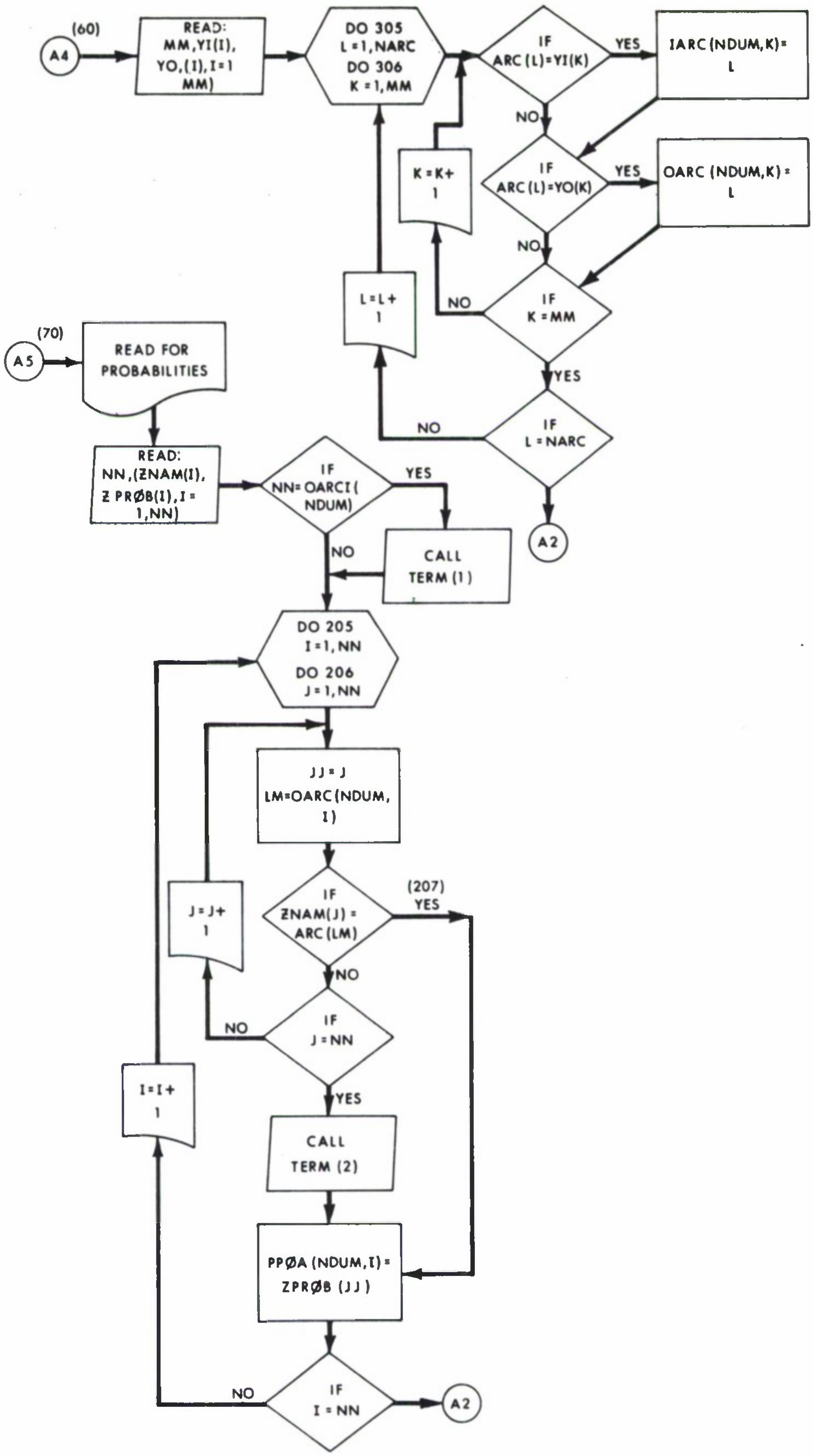
Subroutine IDIN



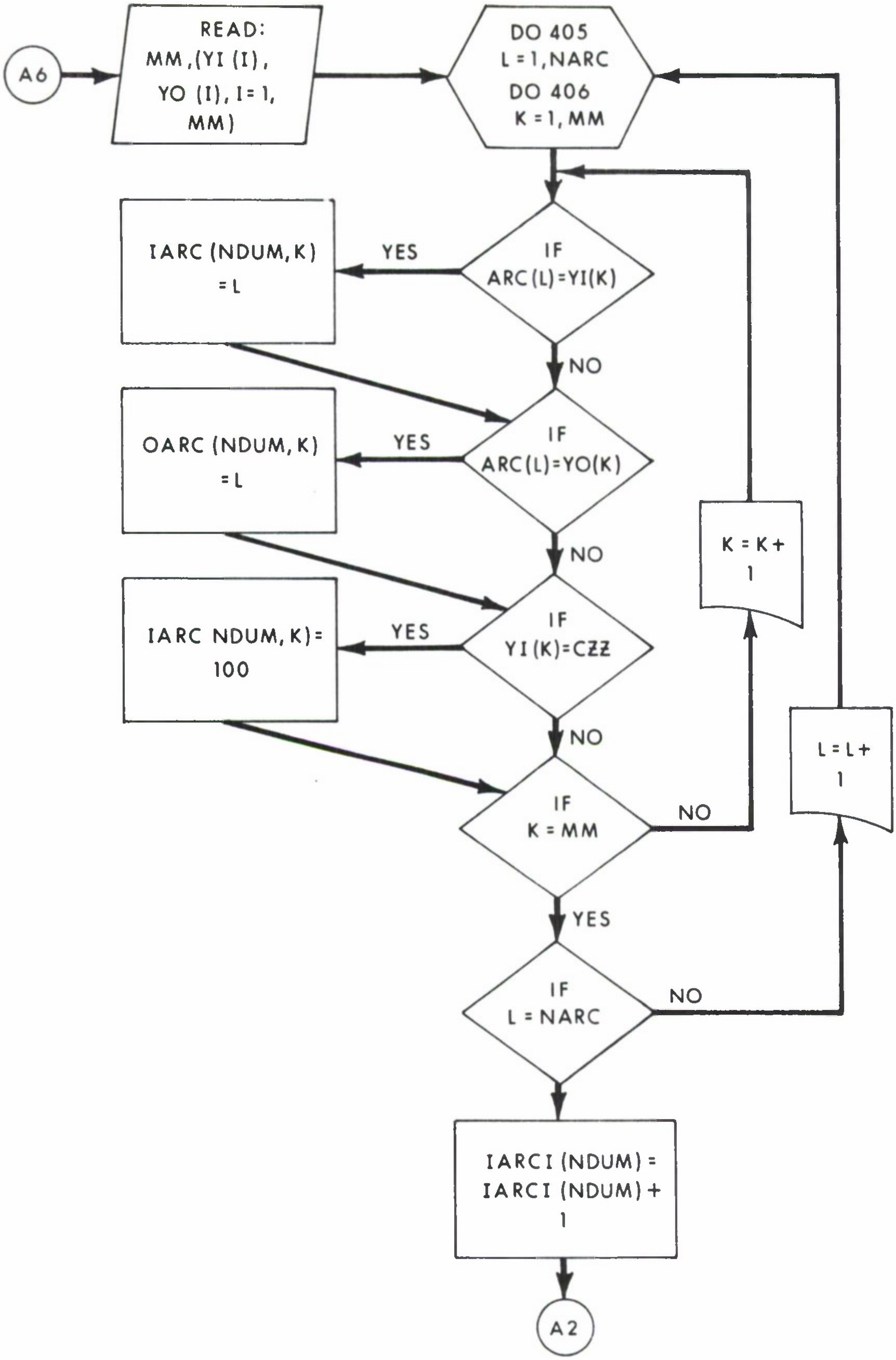
Subroutine ARCIN



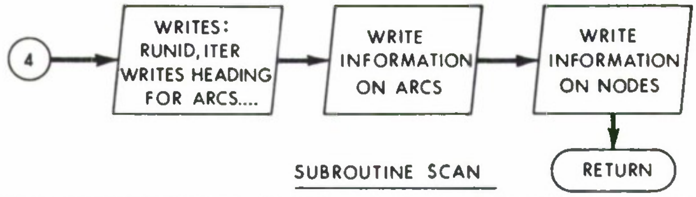
Subroutine NODIN



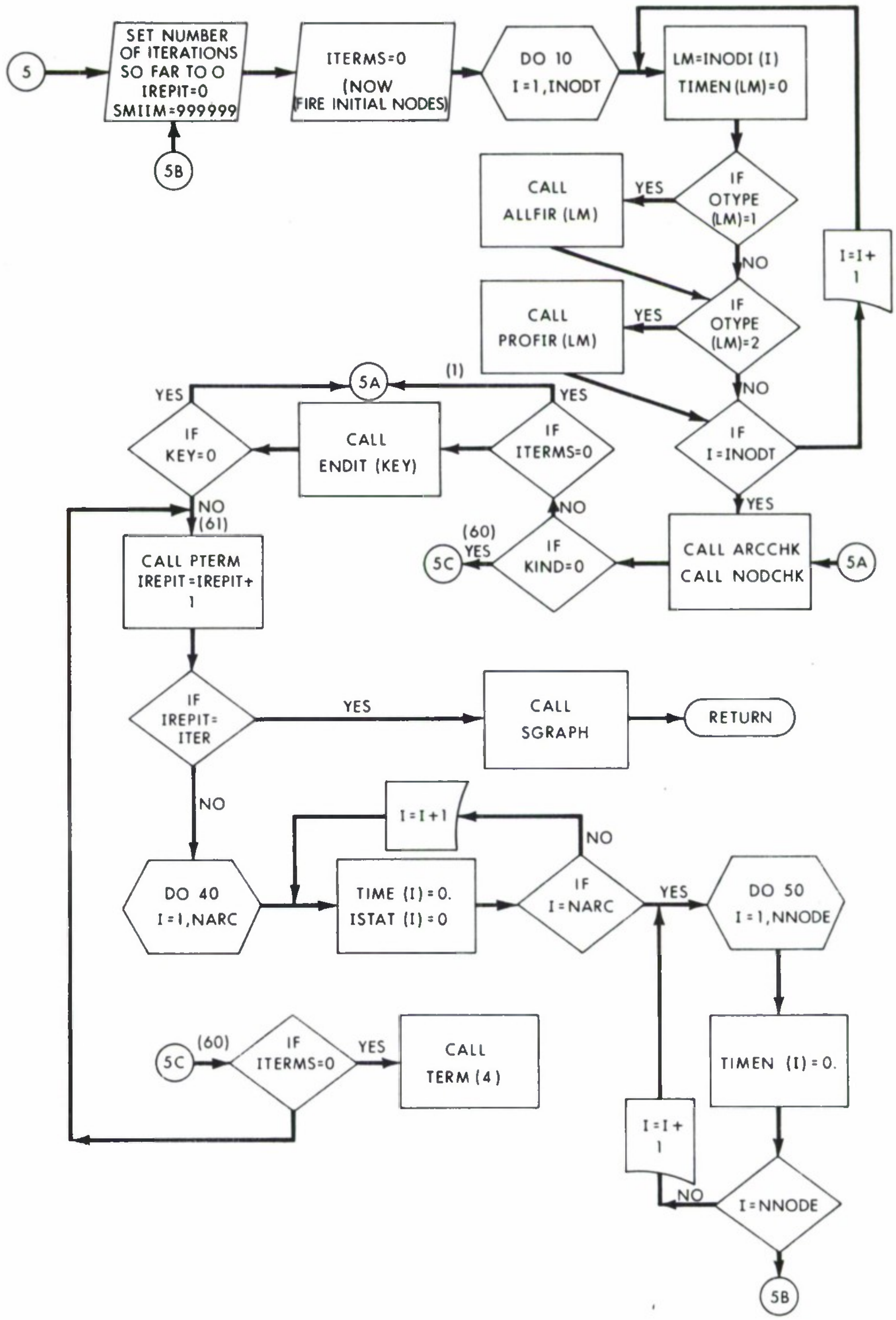
Subroutine NODIN (Continued)



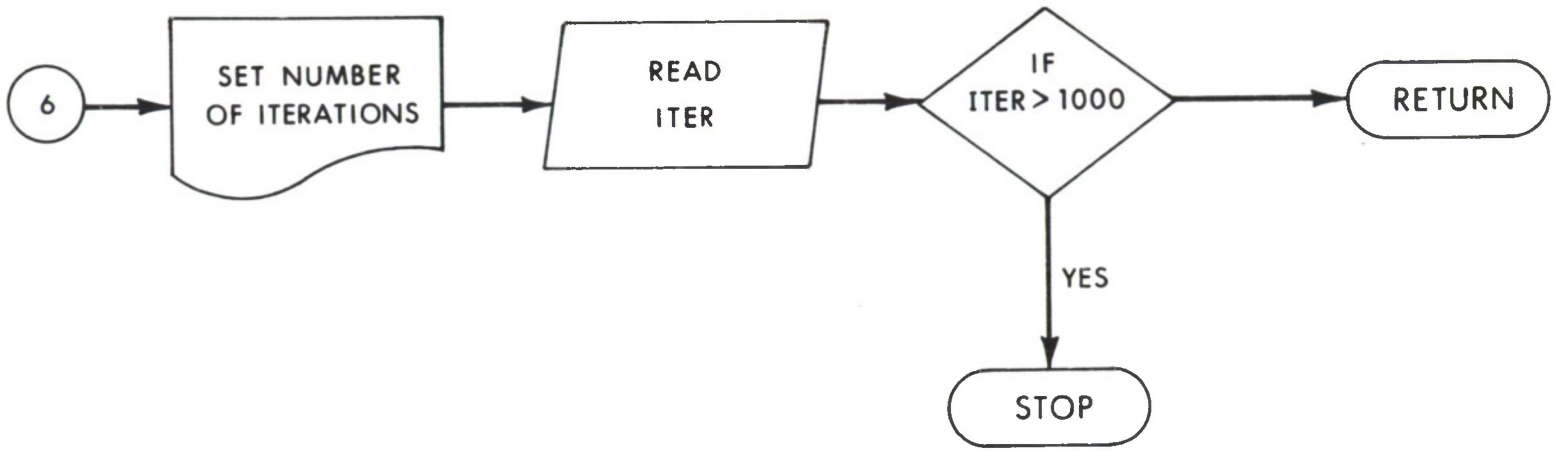
Subroutine NODIN (Continued)



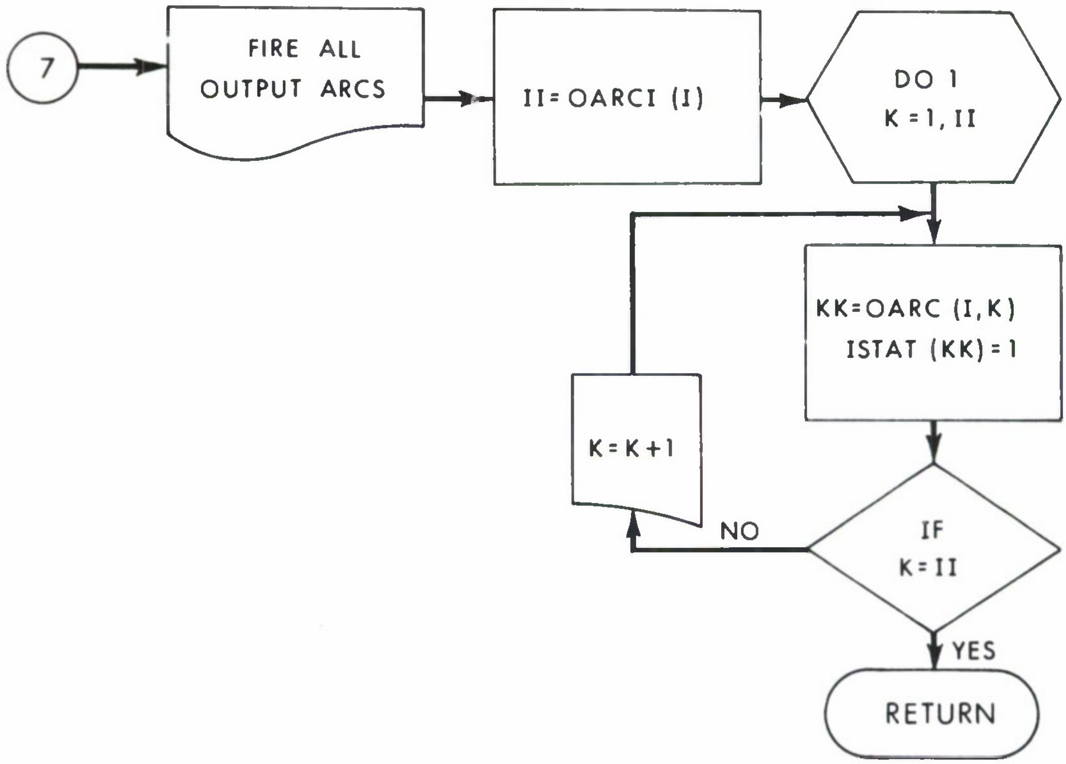
SUBROUTINE SCAN



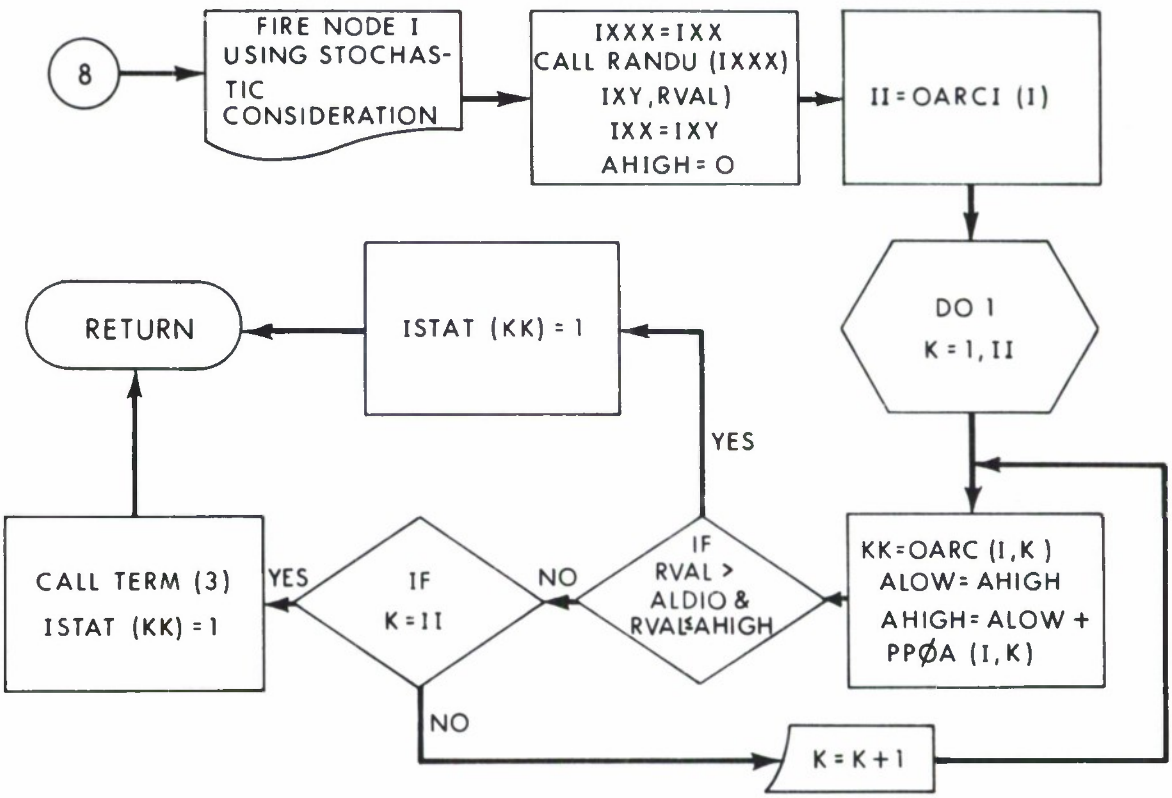
Subroutine RUNSYS



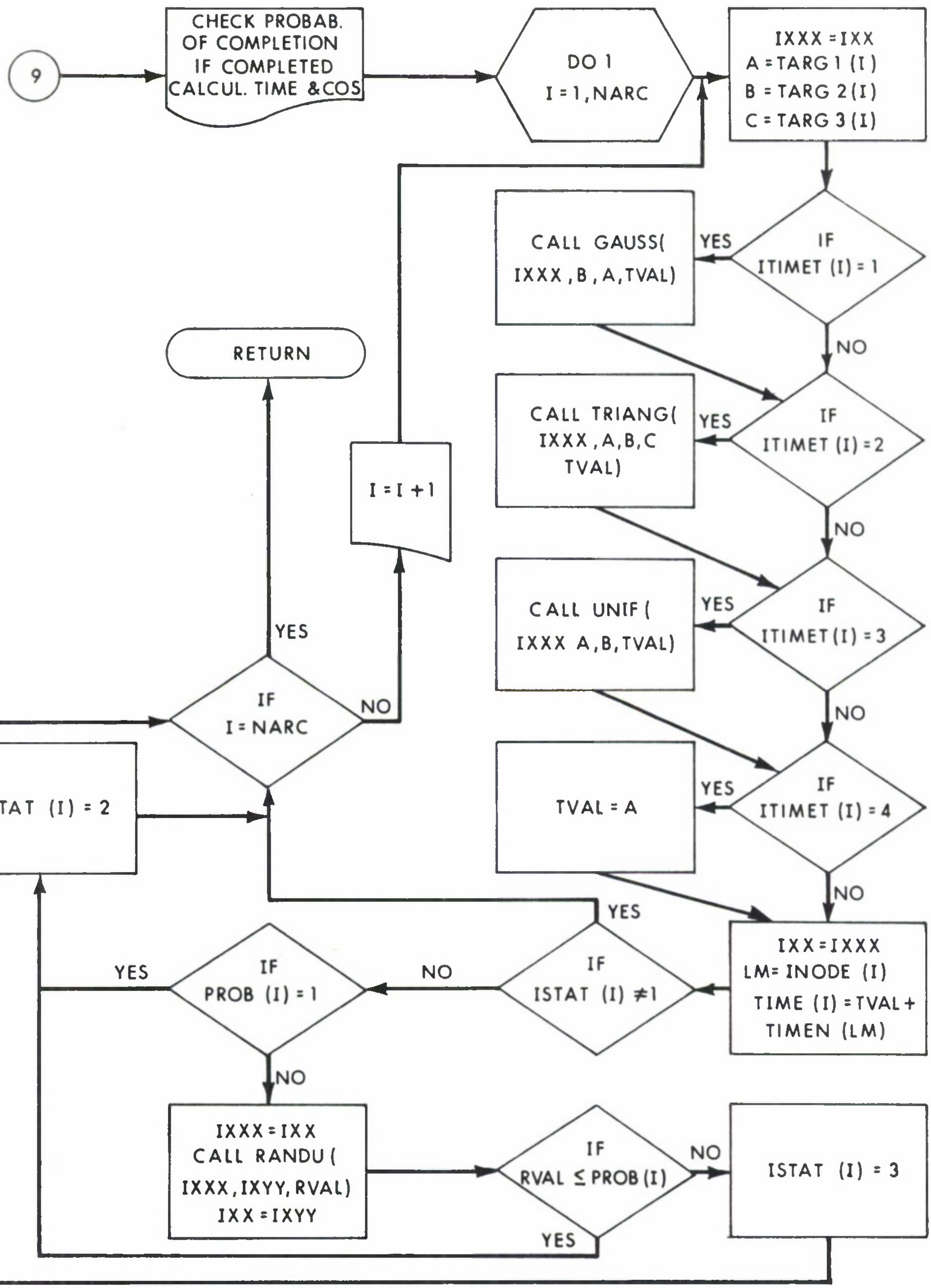
Subroutine REPSET



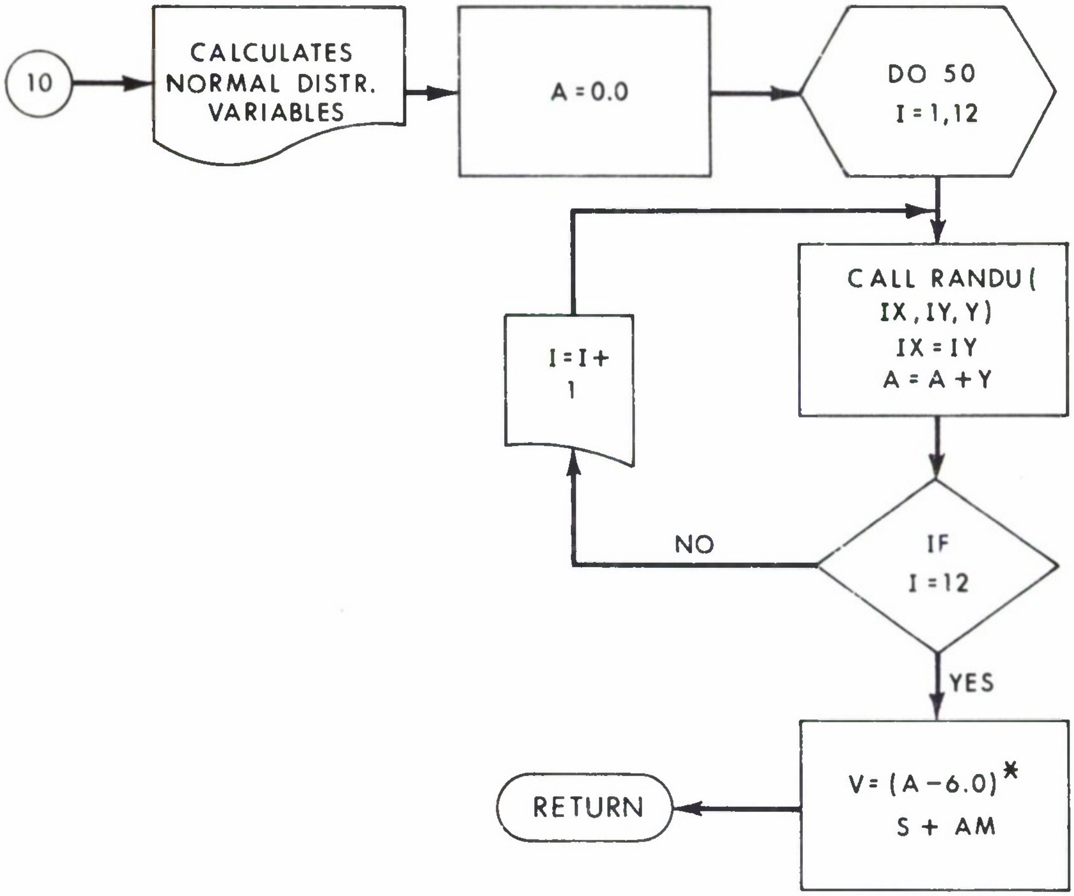
Subroutine ALLFIR (I)



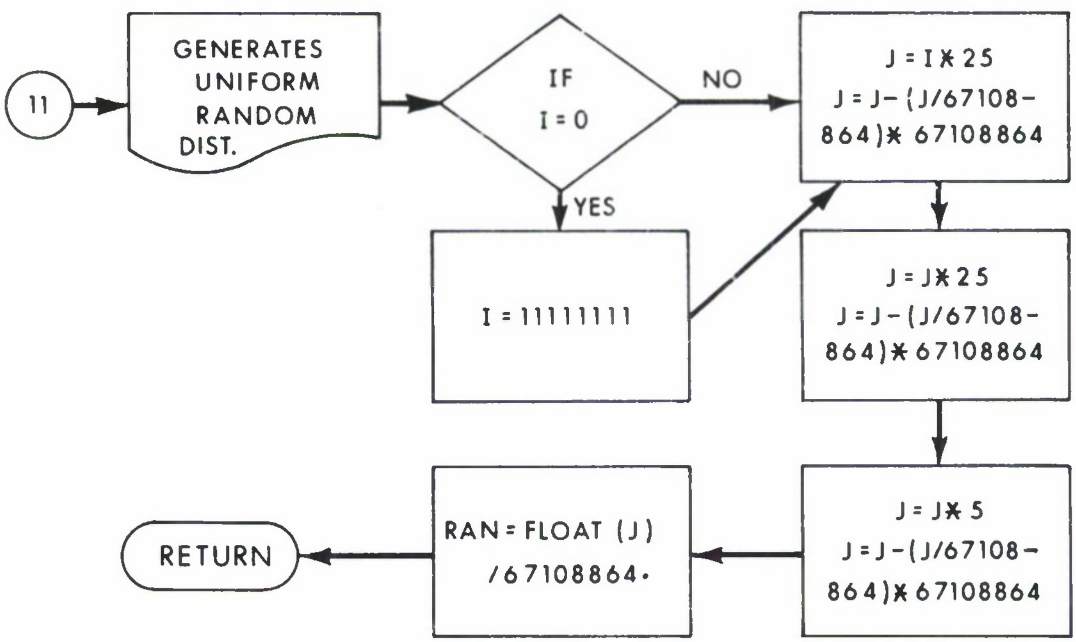
Subroutine PROFIR (I)



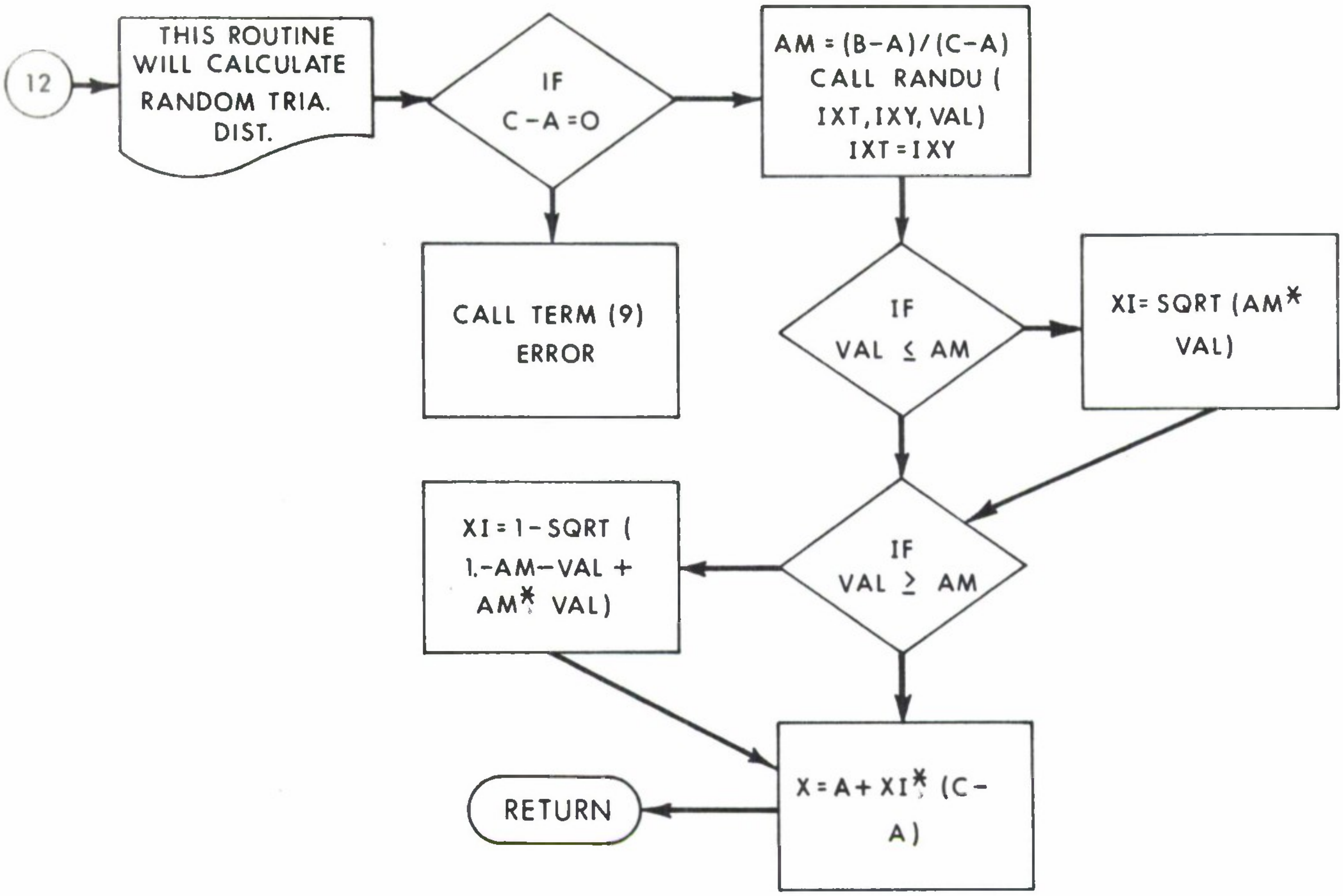
Subroutine ARCCHK



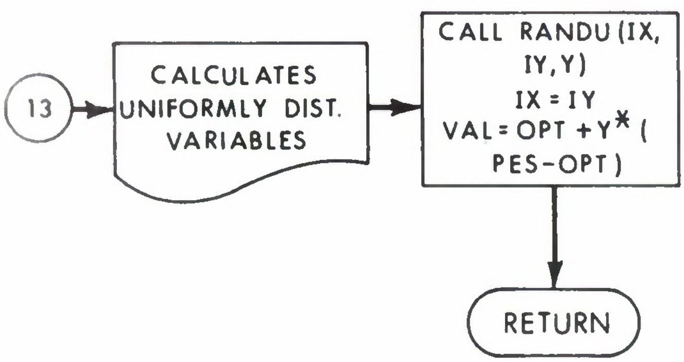
Subroutine GAUSS (IX, S, AN, V)



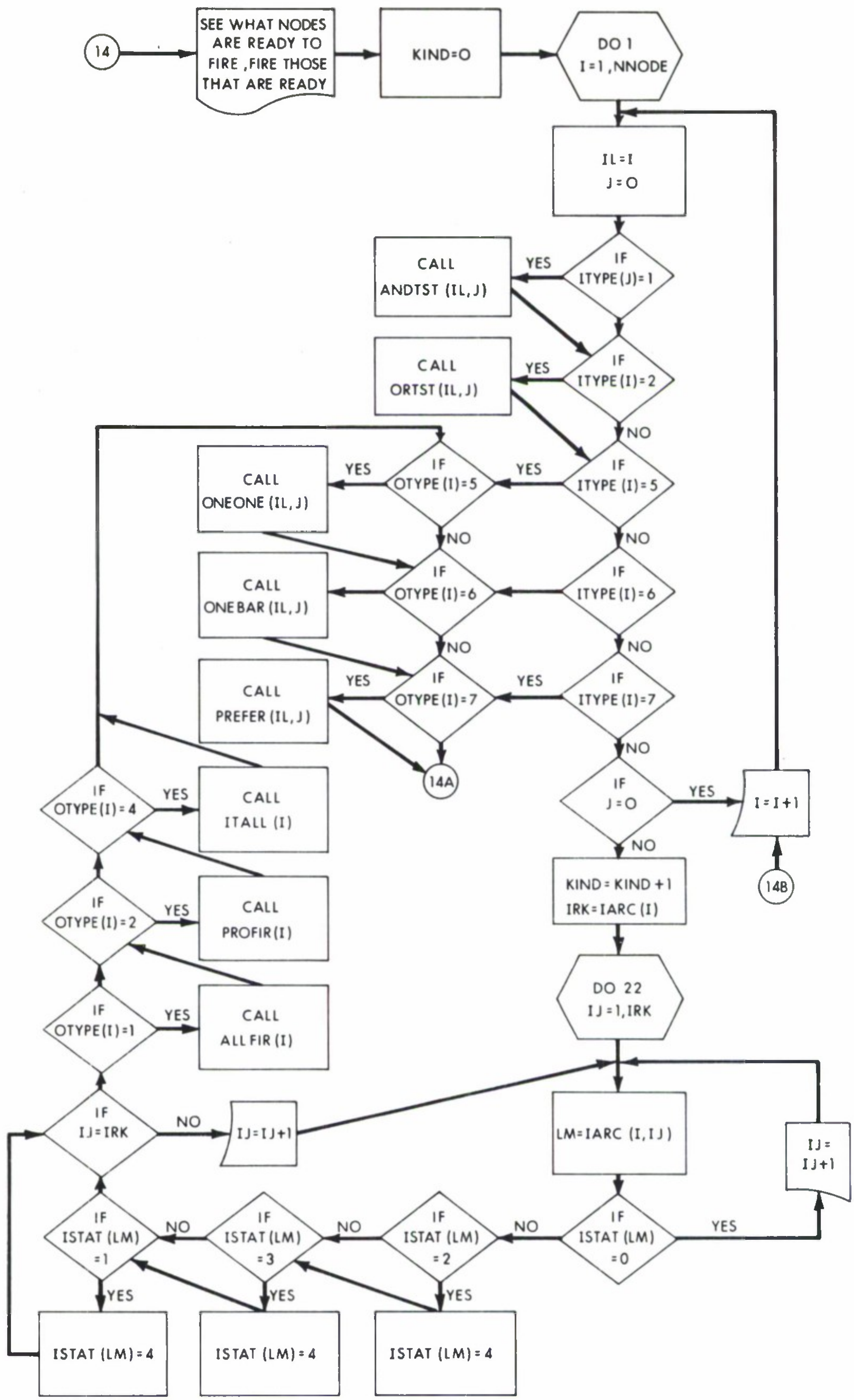
Subroutine RANDU (I, J, RAN)



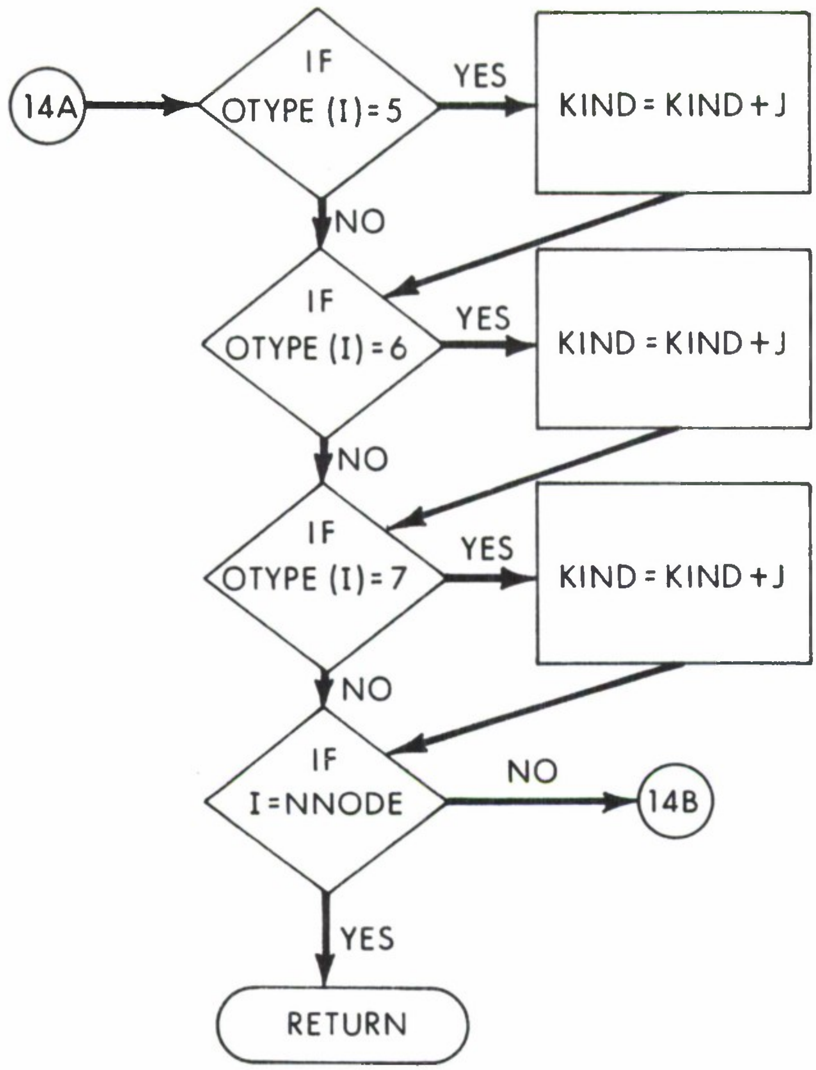
Subroutine TRIANG (IXT. A, B, C, X)



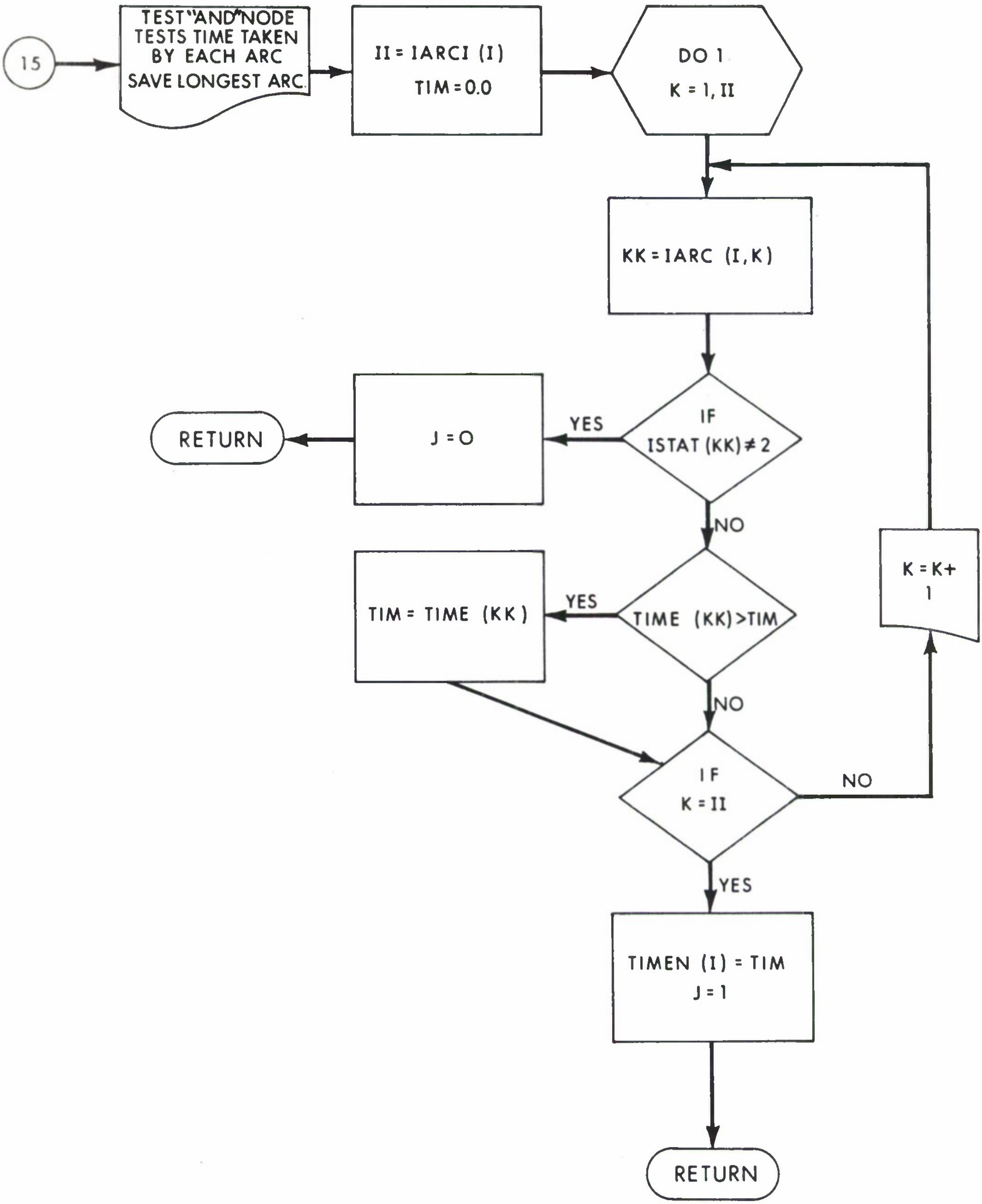
Subroutine UNIF (IX, OPT, PES. VAL)



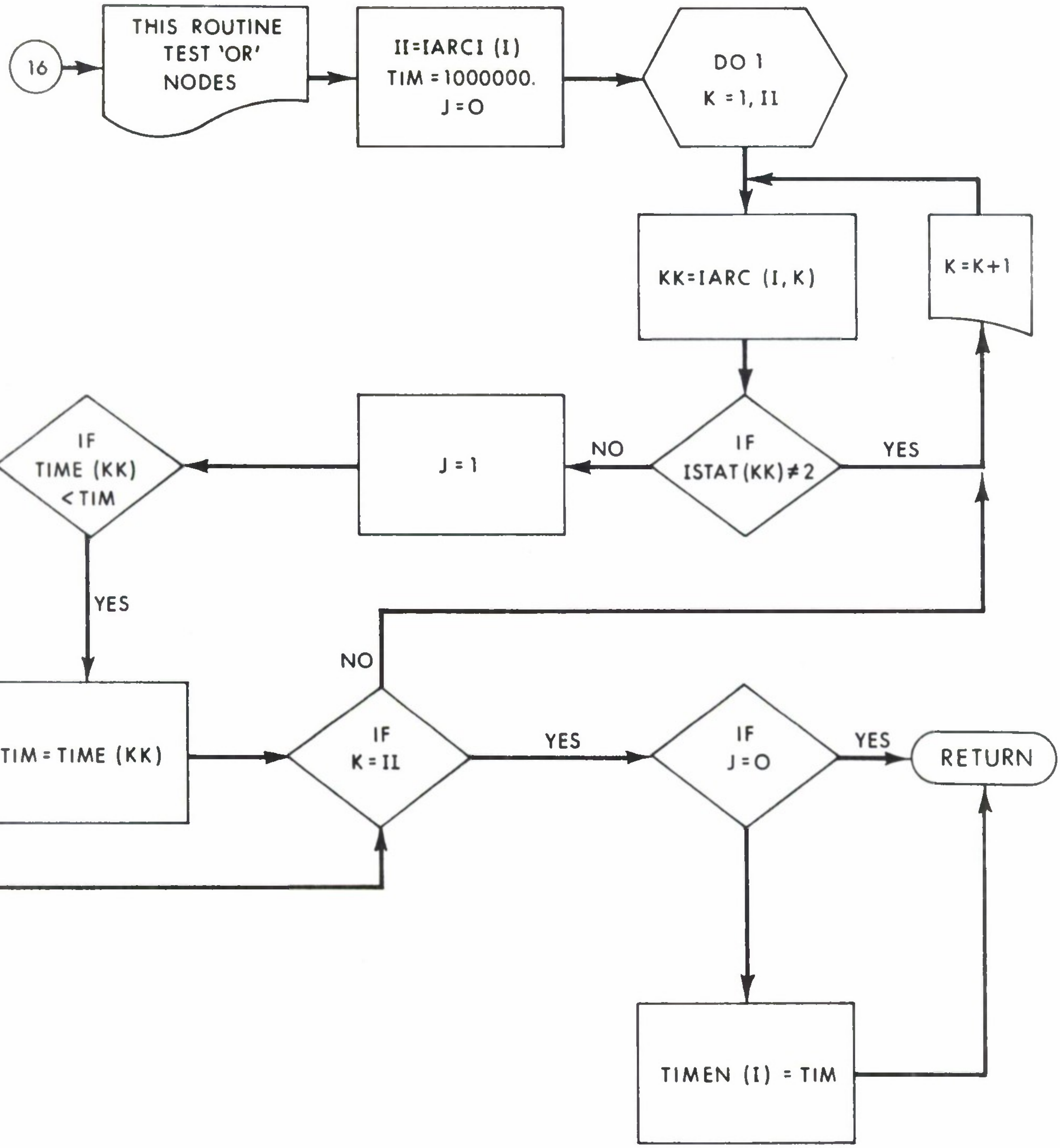
Subroutine NODCHK



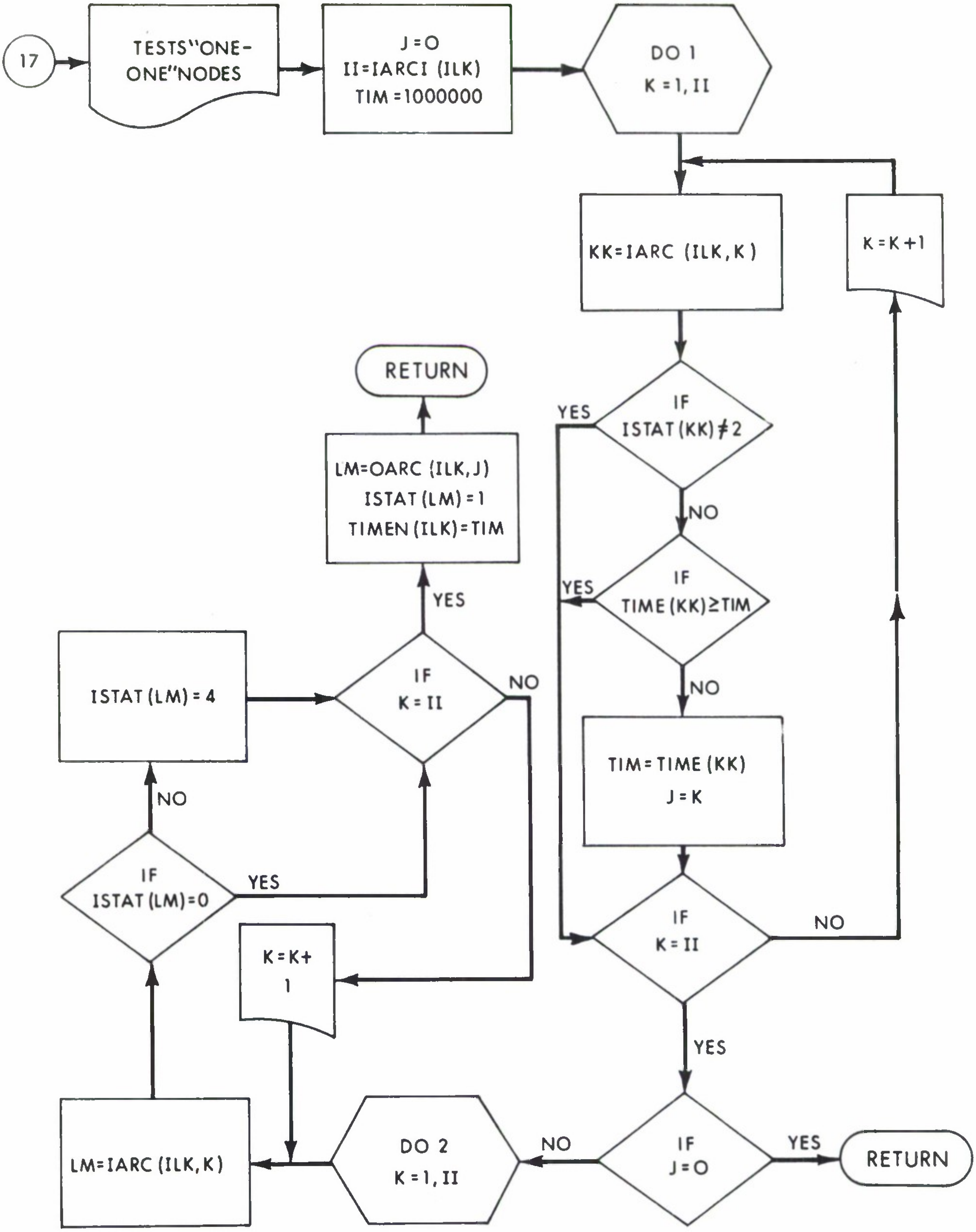
Subroutine NODCHK (Continued)



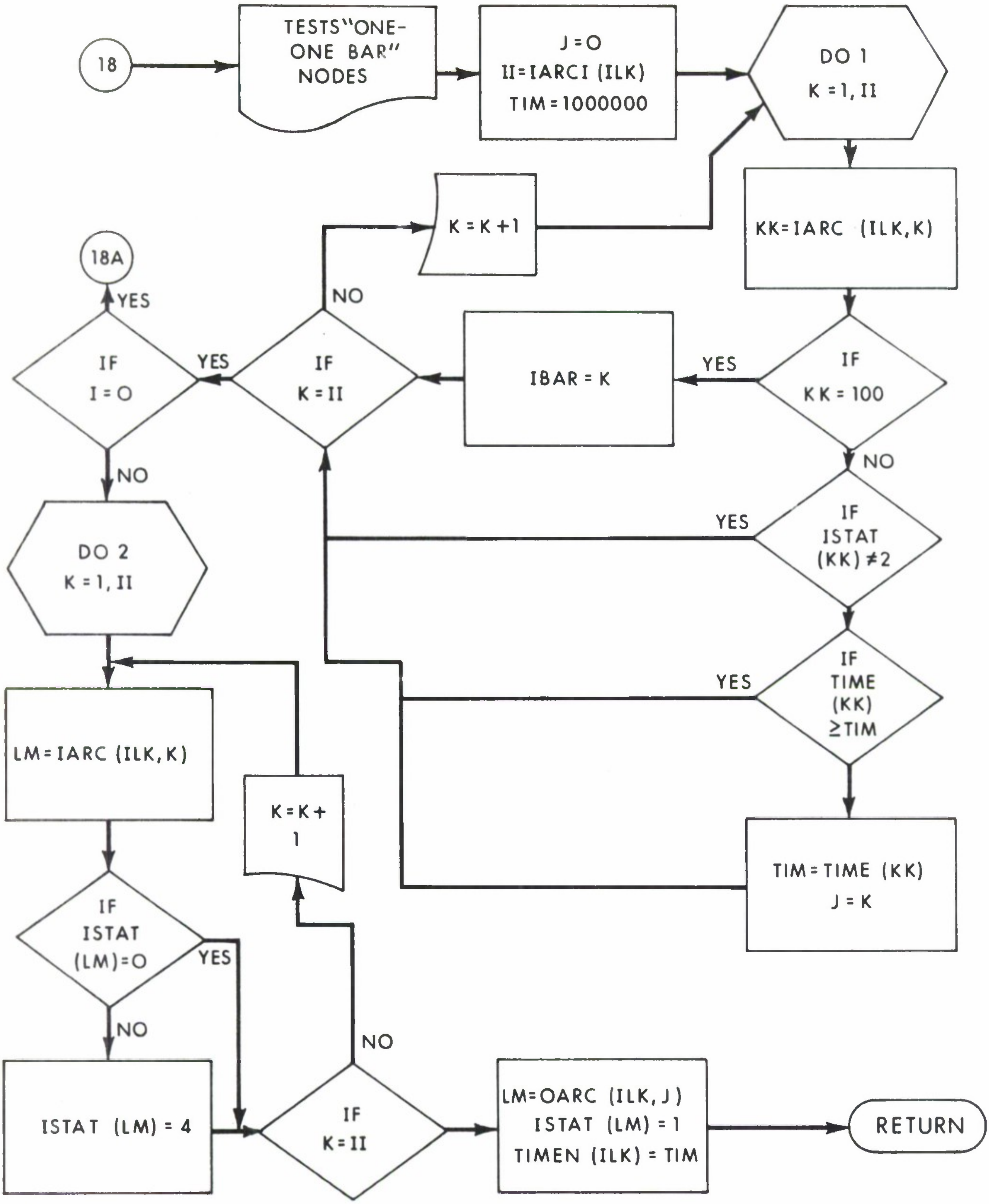
Subroutine ANDTST (I, J)



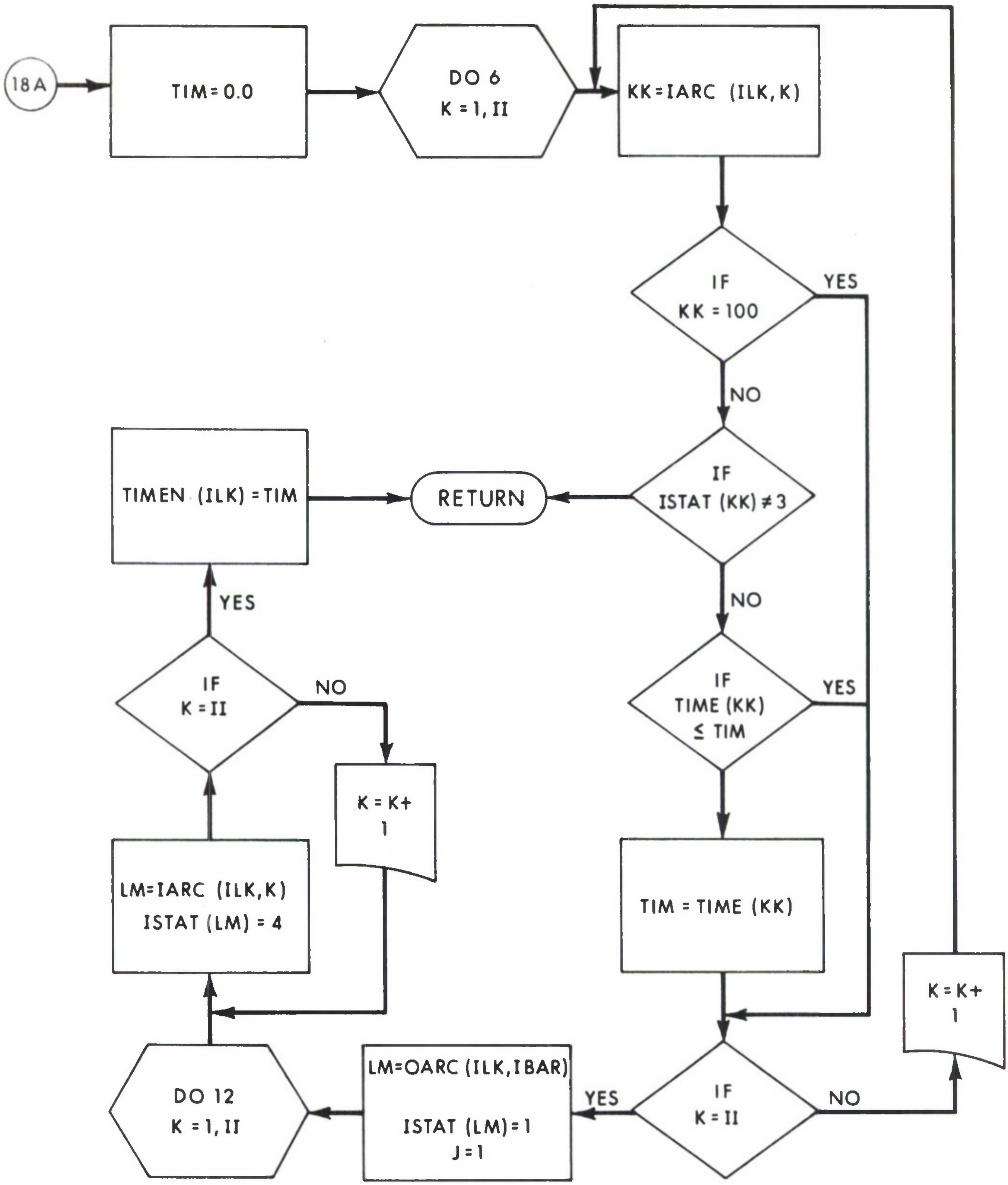
Subroutine ORTST (I, J)



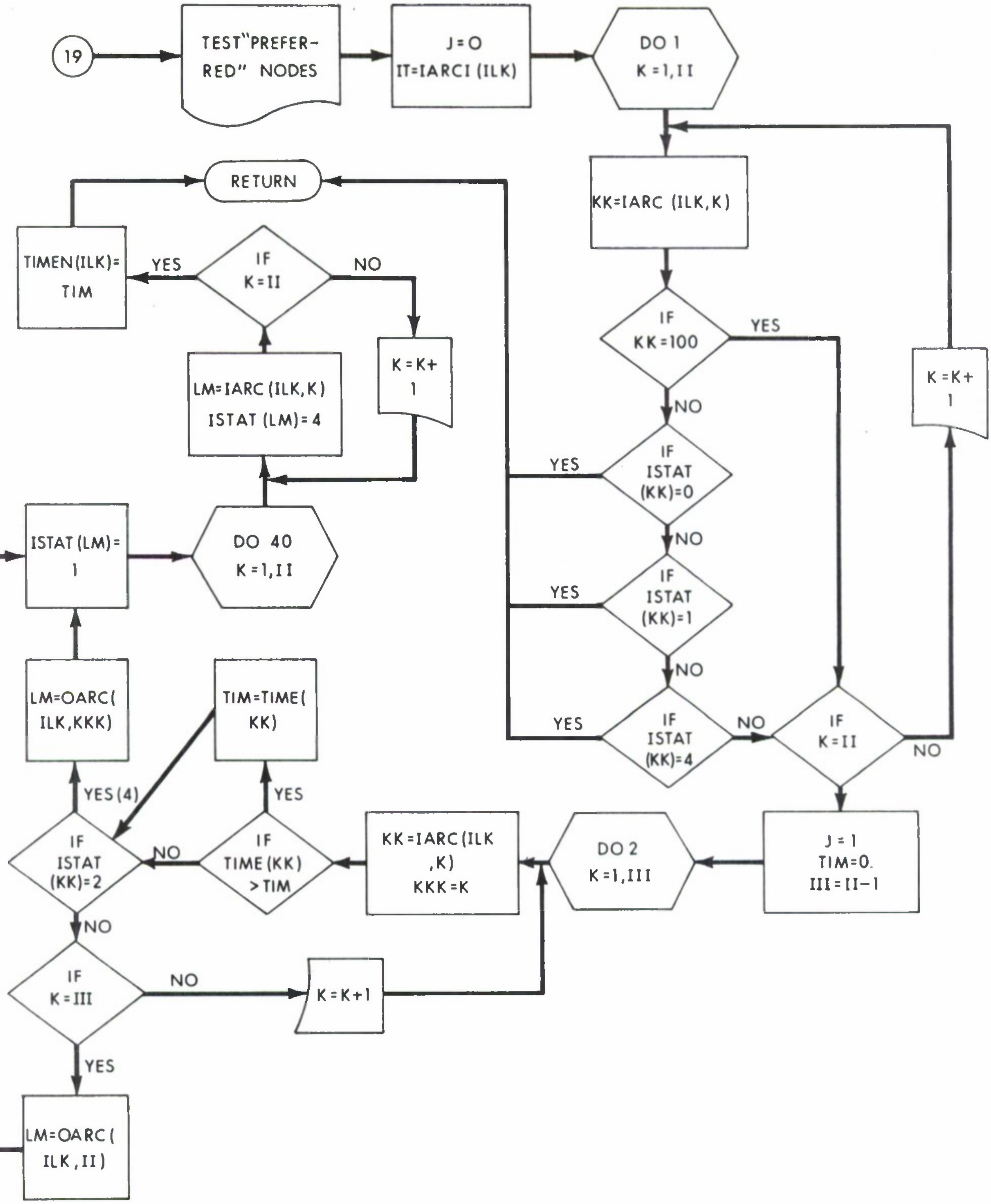
Subroutine ONEONE (ILK, J)



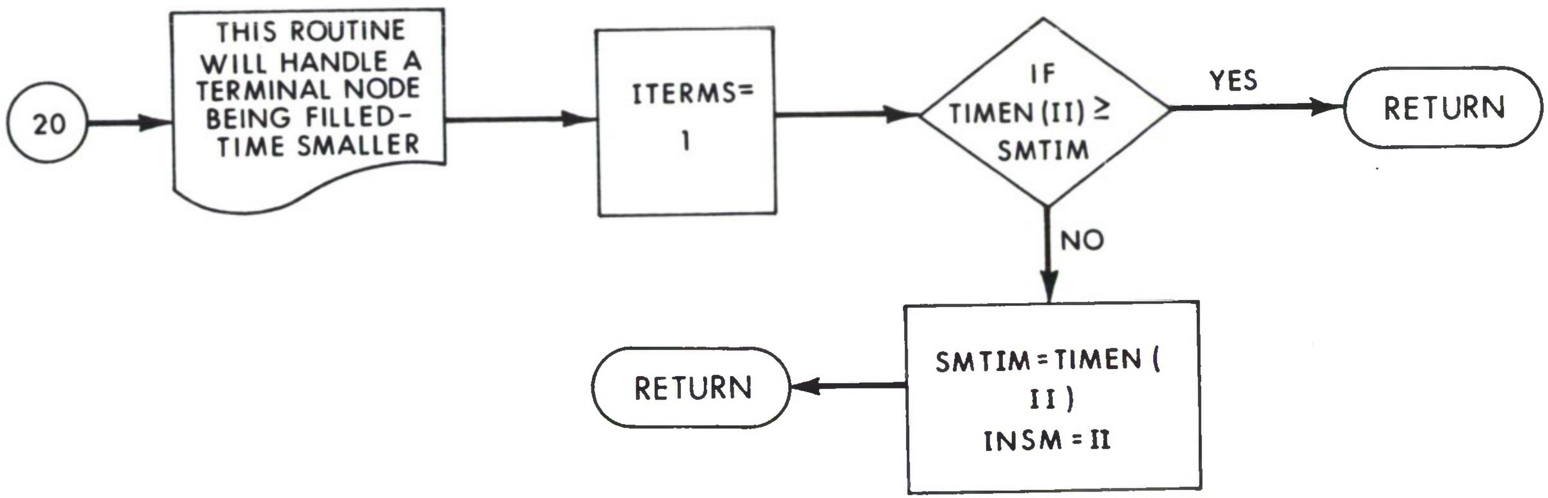
Subroutine ONEBAR (ILK, J)



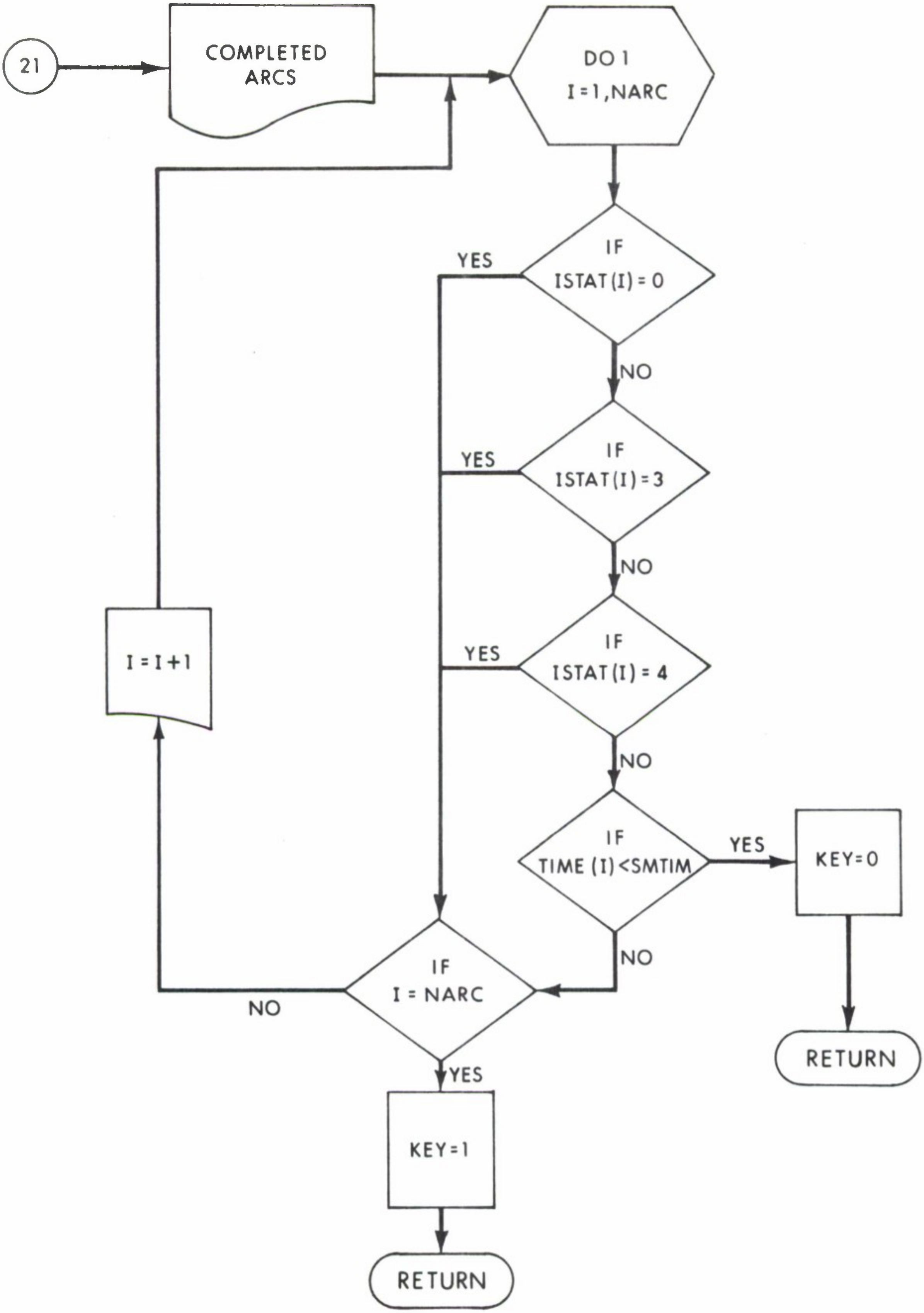
Subroutine ONEBAR (ILK, J) (Continued)



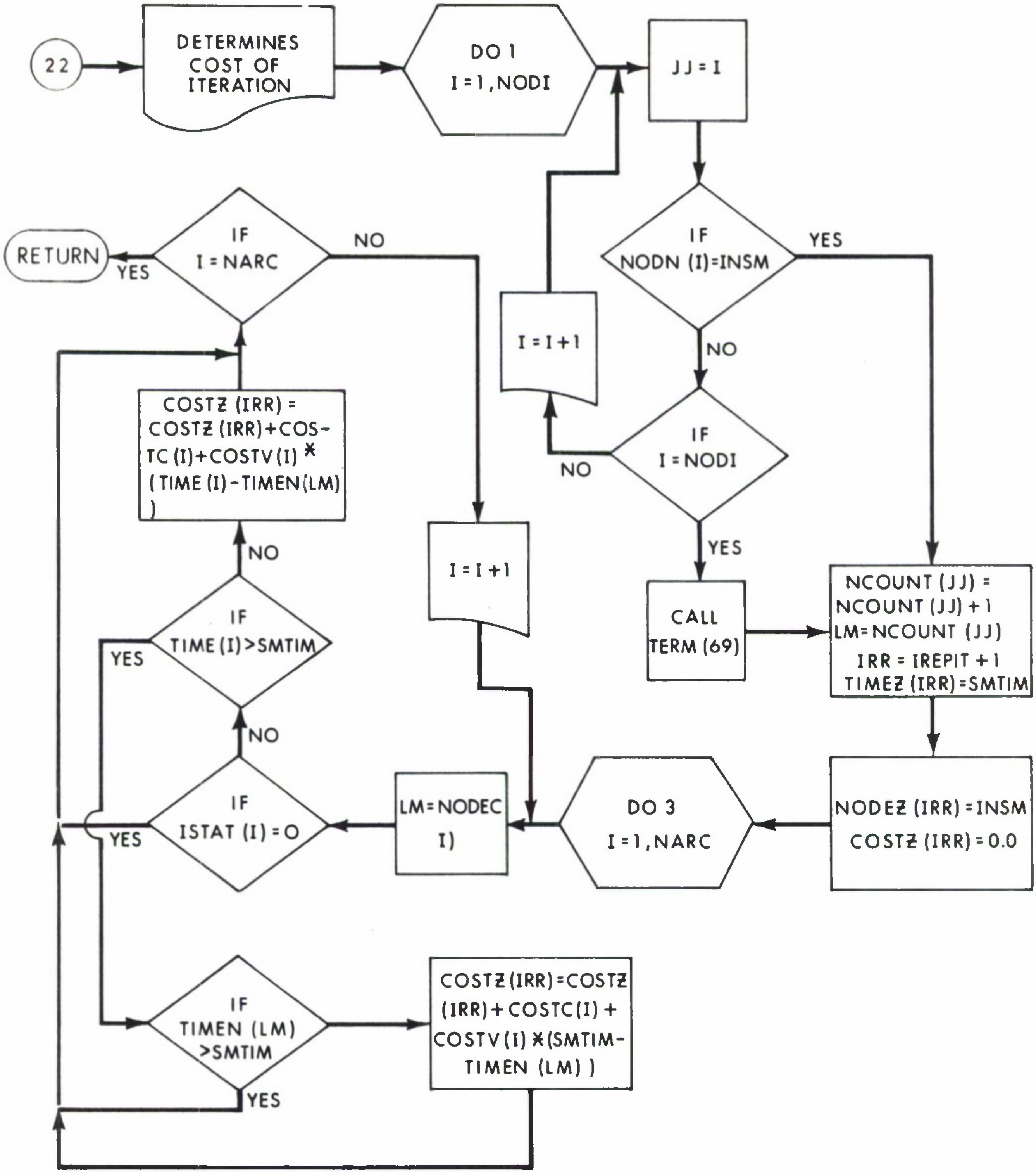
Subroutine PREFER (ILK,J)



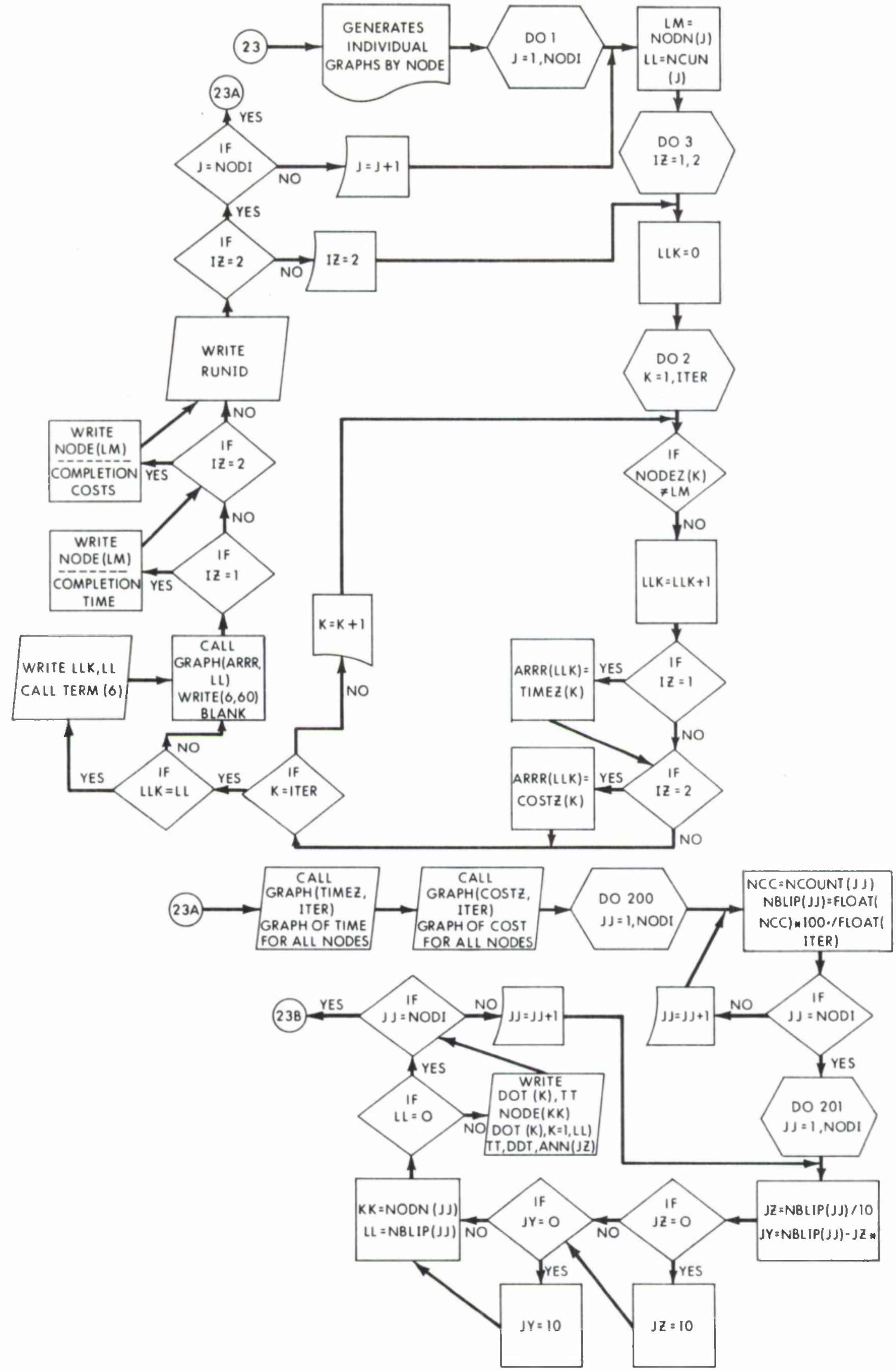
Subroutine ITALL (II)



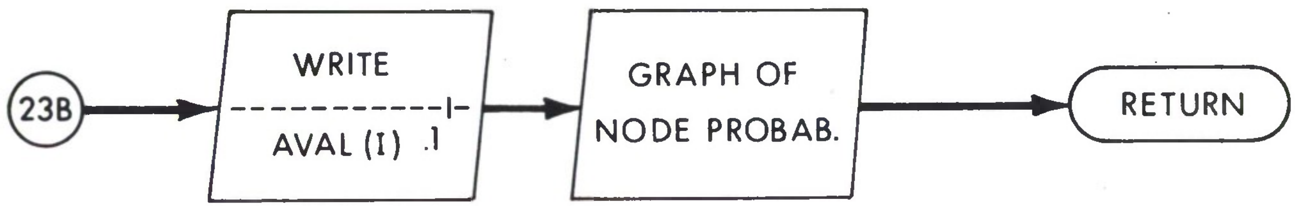
Subroutine ENDIT (KEY)



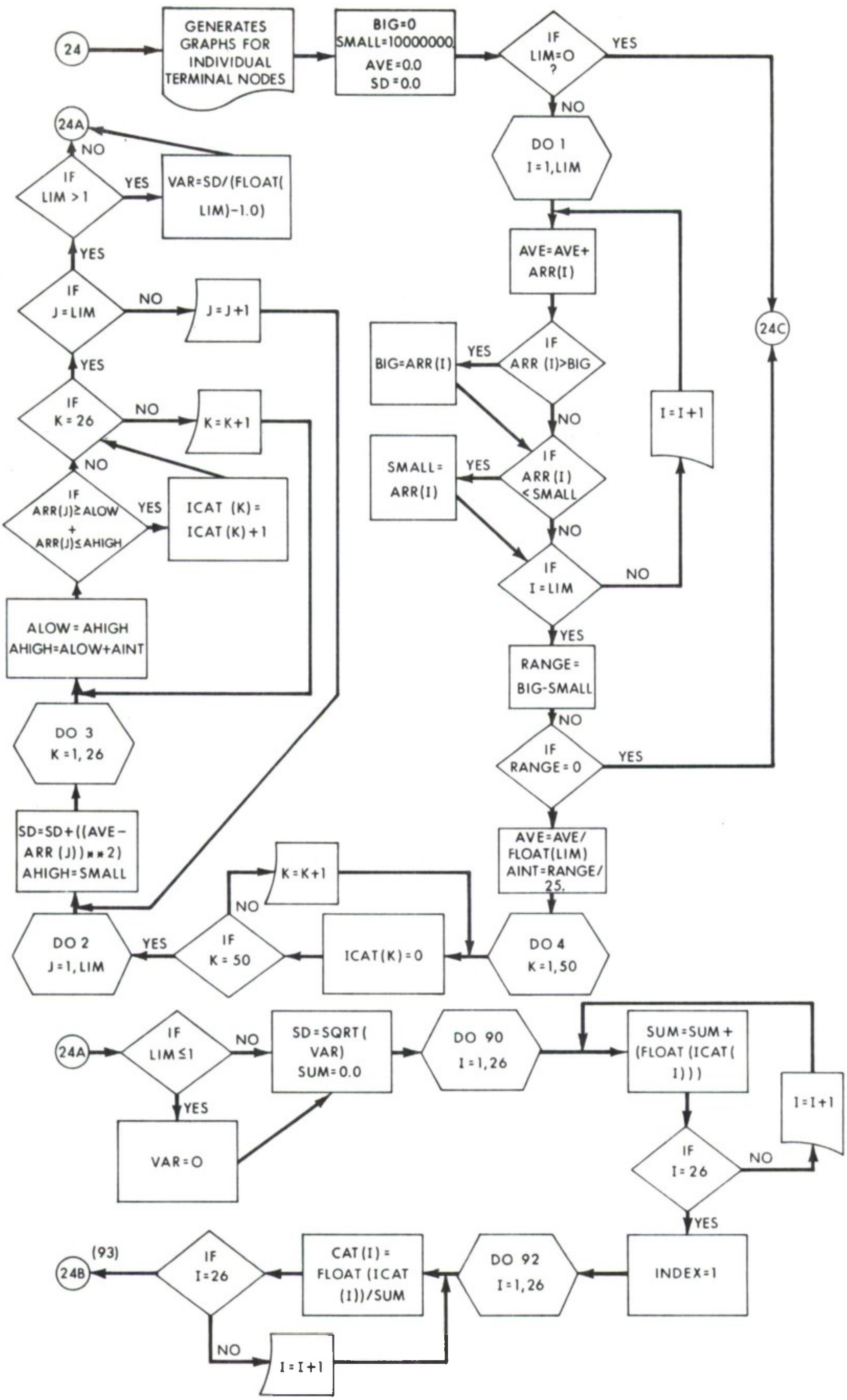
Subroutine PTERM



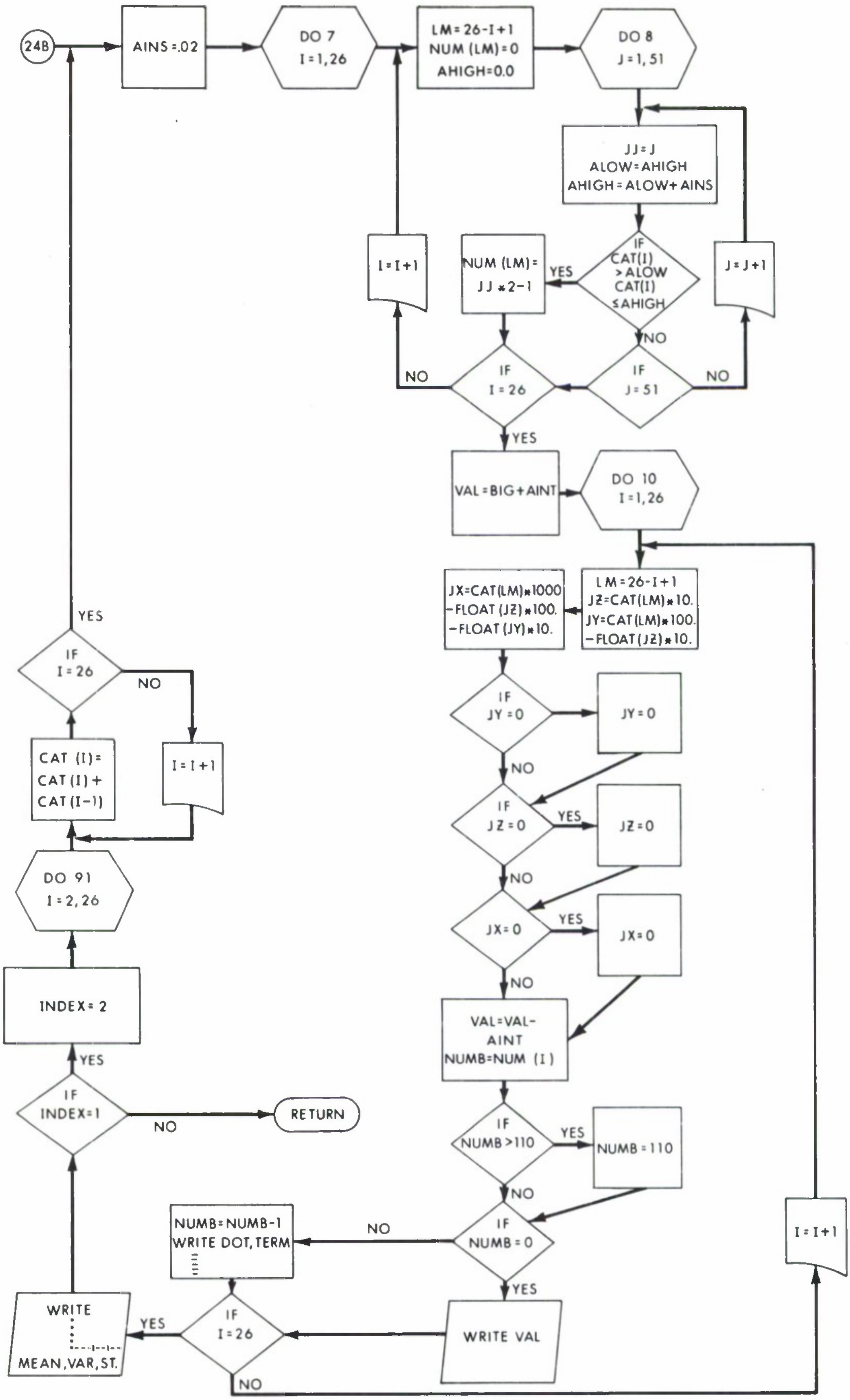
Subroutine SGRAPH



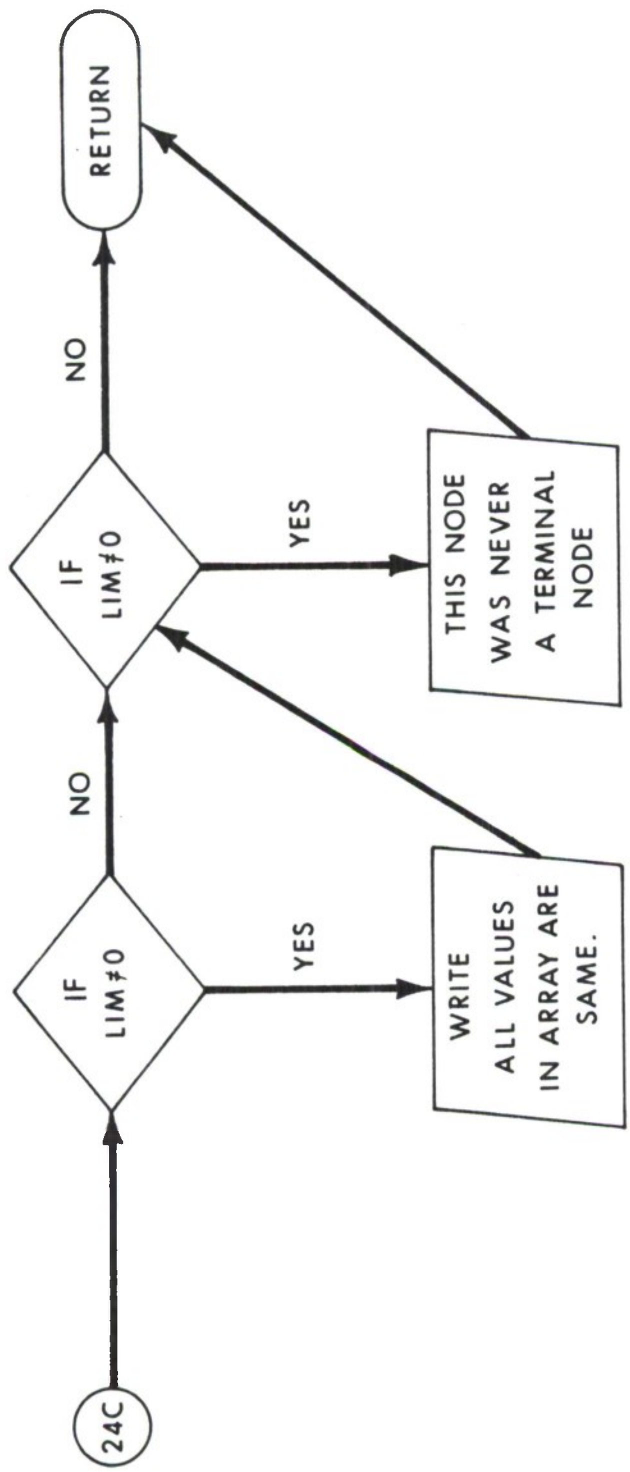
Subroutine SGRAPH (Continued)

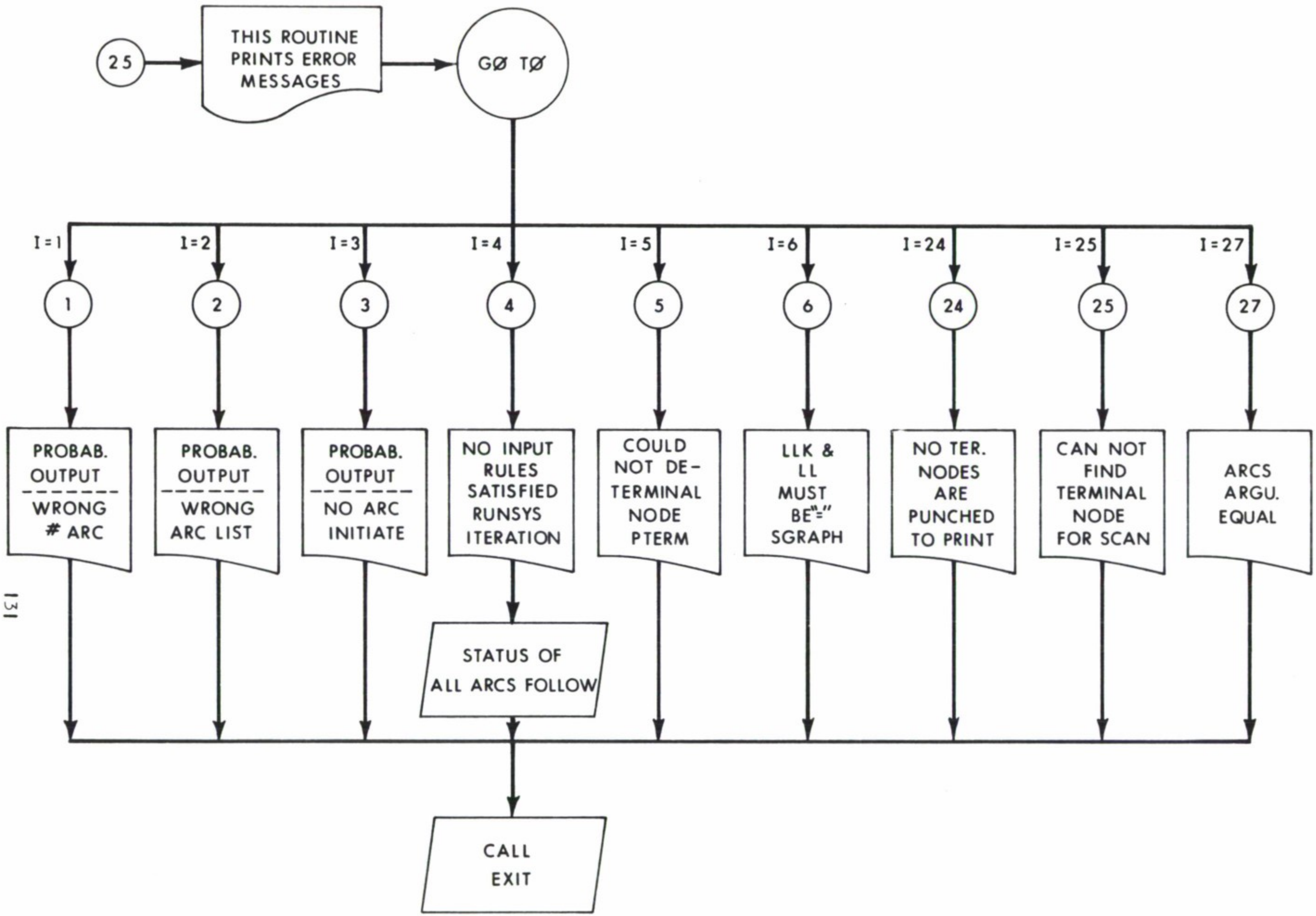


Subroutine GRAPH (ARR,LIM)



24B
AINS =.02
DO 7
I = 1, 26
LM = 26-I+1
NUM (LM) = 0
AHIGH=0.0
DO 8
J = 1, 51
JJ = J
ALOW = AHIGH
AHIGH = ALOW+ AINS
IF
CAT(I)
> ALOW
CAT(I)
≤AHIGH
YES
NUM (LM) =
JJ * 2-1
J = J+1
I = I+1
NO
IF
J = 51
NO
IF
I = 26
NO
YES
VAL = BIG + AINT
DO 10
I = 1,26
LM = 26-I+1
JZ=CAT(LM)*10.
JY=CAT(LM)*100.
-FLOAT(JZ)*10.
JX=CAT(LM)*1000
-FLOAT (JZ)*100.
-FLOAT (JY)*10.
IF
JY = 0
JY = 0
NO
IF
JZ = 0
YES
JZ = 0
NO
JX = 0
YES
JX = 0
NO
VAL=VAL-
AINT
NUMB=NUM (I)
IF
NUMB >110
YES
NUMB = 110
NO
IF
NUMB = 0
NO
NUMB=NUMB-1
WRITE DOT, TERM
YES
WRITE VAL
IF
I = 26
NO
I = I+1
YES
WRITE
MEAN, VAR, ST.
IF
INDEX=1
NO
RETURN
YES
INDEX = 2
DO 91
I = 2, 26
CAT (I) =
CAT (I) +
CAT (I-1)
IF
I = 26
NO
I = I+1
YES


24C
IF
LIM≠0
YES
WRITE
ALL VALUES
IN ARRAY ARE
SAME.
NO
IF
LIM≠0
YES
THIS NODE
WAS NEVER
A TERMINAL
NODE
NO
RETURN



Subroutine TERM (I)

In MATHNET, the error message refers to an integer I and is stated:

"EXECUTION TERMINATED FOR REASON I"

The following is a list of the possible integers and the errors they indicate:

a. I = 99. the number of stochastic outputs from node x (x is stated) is not correct - (NODIN)

b. I = 3030. no input rules were satisfied (RUNSYS)

c. I = 1210. I/I output is not correct (NODCHK)

d. I = 1211. $\overline{I}$/I output is not correct (NODCHK)

e. I = 1212. preferred output is not correct (NODCHK)

f. I = 100. probabilistic output node - wrong arc listed (NODIN)

g. I = 69. probabilistic output node - wrong number of arcs stated (PTERN)

h. I = 1. probabilistic output node - no arc initiated (PROFIR)

i. I - 444.(6) number of iterations set is greater than 1000 (SGRAPH).

In RISCA, the errors which result in termination of the execution are printed as statements.

The following is a list of the RISCA error statements:

a. ***ERROR*** PROBABILISTIC OUTPUT NODE - WRONG NUMBER OF ARCS STATED***NODIN

b. ***ERROR***PROBABILISTIC OUTPUT NODE - WRONG ARC LISTED*** NODIN

c. ***ERROR***PROBABILISTIC OUTPUT NODE - NO ARC INITIATED***PROFIR

d. ***ERROR***NO INPUT RULES WERE SATISFIED***RUNSYS

e. ***ERROR***COULD NOT DETERMINE TERMINAL NODE***PTERM

f. ***ERROR***LLK AND LL MUST BE EQUAL***SGRAPH

g. ***ERROR***NO TERMINAL NODES ARE PUNCHED TO PRINT

h. ***ERROR***CAN NOT FIND TERMINAL NODES FOR SCAN

i. ***ERROR***CHECK ALL ARC CARDS AT LEAST ONE SHOWS FIRST AND THIRD ARGUMENT EQUAL WHILE TIME DISTRIBUTION TYPE IS TRIANGULAR CHANGE TO CONSTANT.

Next page is blank.

APPENDIX IV

MATHNET OUTPUT FOR THE GOING TO WORK PROBLEM

Next page is blank.

```
IF YOU ARE RUNNING THIS FROM A TERMINAL
PLEASE ENTER A 1, IF RUNNING BATCH YOU SHOULD
HAVE ENTERED A CARD WITH A 0
FORMAT IS I1
YOU ARE NOW IN MONITOR MOOE , FROM THE FOLLOWING LIST
SELECT THE MODE YOU WISH TO GO INTO
1     ENTER NODES
2     ENTER ARCS
3     SET ITERATION NUMBER
4     SCAN THE NET SO FAR
5     RUN NET
6     ENTER RUN IDENTIFIER
9     END SESSION
*
 ENTER A RUN IOENTIFIER OF 80 CHARACTERS OR LESS
*
 YOU HAVE RETURNED TO MONITOR MODE
 SELECT THE MOOE YOU WISH TO GO INTO AS INDICATED BEFORE
*
 YOU CAN NOW SET THE NUMBER OF ITERATIONS
 ENTER A 5 POSITION INTEGER, RIGHT AOJUSTEO
 *
*
 YOU HAVE RETURNED TO MONITOR MOOE
 SELECT THE MOOE YOU WISH TO GO INTO AS INOICATEO BEFORE
*
 YOU ARE NOW IN ENTER ARC MOOE
*
 ENTER ARC NAME,INPUT NODE NAME,OUTPUT NODE NAME,
 TIME OISTRIBUTION TYPE,TIME DISTRIBUTION ARGUMENTS 1,2,3,
 CONSTANT COST COEFFICIENT,COEFFICIENT OF TIME TERM IN        COST TERM,

 PROBABILITY OF SUCCESSFUL ARC COMPLETION
 FORMAT IS 3A4,I1,6F10.0
*
 TO RETURN TO MONITOR MODE ENTER  RETU
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
```

```
*
*
*
 YOU HAVE RETURNED TO MONITOR MOOE
 SELECT THE MOOE YOU WISH TO GO INTO AS INOICATEO BEFORE
*
 YOU ARE IN ENTER NOOE MOOE
 ENTER NODE NAME,INPUT RULE,OUTPUT RULE
 FORMAT IS A4,I1,I1
 INPUT AND OUTPUT RULES ARE AS FOLLOWS
           RULE NUMBER           INPUT RULE     OUTPUT RULE
          .............         ............              .............
                1                   ANO                        ALL FIRE
                2                   OR                         PROB. FIRE
                4                   INITIAL                    TERMINAL
                5                   1/1                        1/1
                6                   1/1 BAR                    1/1 BAR
                7                   PREFERREO                  PREFERREO
*
*
 YOU HAVE INOICATEO A NODE WITH STOCHASTIC OUTPUTS
 INPUT NUMBER OF OUTPUT ARCS NAME OF OUTPUT ARC,PROB..
 FORMAT IS I2,10(A4,F6.3)
*
*
 YOU HAVE INOICATEO A 1/1 NOOE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAMES
 FORMAT IS I2,10(A4,A4
*
*
*
 YOU HAVE INOICATEO A 1/1 BAR NODE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
 INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
 FORMAT IS I2,10(A4,A4
*
*
 YOU HAVE INOICATED A 1/1 BAR NOOE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
 INPUT A NAME OF ZZZZ FOR THE NO INPUT CONOITION
 FORMAT IS I2,10(A4,A4
*
*
 YOU HAVE INOICATEO A 1/1 BAR NOOE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
 INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
 FORMAT IS I2,10(A4,A4
*
*
 YOU HAVE INDICATED A 1/1 BAR NOOE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
 INPUT A NAME OF ZZZZ FOR THE NO INPUT CONOITION
 FORMAT IS I2,10(A4,A4
*
*
*
*
 YOU HAVE INDICATED A 1/1 BAR NODE
 INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
 INPUT A NAME OF ZZZZ FOR THE NO INPUT CONOITION
```

```
FORMAT IS I2,10(A4,A4
*
*
YOU HAVE INDICATED A 1/1 BAR NODE
INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,10(A4,A4
*
*
YOU HAVE INDICATEO A 1/1 BAR NOOE
INPUT NUMBER OF ARCS INPUT ARC NAME,OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,10(A4,A4
*
*
*
*
*
*
*
*
*
*
YOU HAVE RETURNED TO MONITOR MOOE
SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE
*
```

ARC	INPUT NODE	OUTPUT NOOE	TIME DIST	ARG1	ARG2	ARG3	COST		PROB
ARC1	NOD1	NOD2	2	10.00	12.00	14.00	0.00 +	0.00	1.000
ARC2	NOD2	NOO3	2	5.00	6.00	7.00	0.00 +	0.00	1.000
ARC3	NOO2	NOD3	2	11.00	13.00	15.00	0.00 +	0.00	1.000
ARC4	NOD3	NOD4	3	0.00	0.00	0.00	0.00 +	0.00	1.000
ARC5	NOD3	NOD5	3	0.00	0.00	0.00	0.00 +	0.00	0.800
ARC6	NOD4	NOD6	2	3.00	4.00	5.00	0.00 +	0.00	0.900
ARC7	NOD5	NOD7	2	3.00	4.00	5.00	0.00 +	0.00	0.900
ARC8	NOO5	NOO9	3	5.00	5.00	5.00	0.00 +	0.00	1.000
ARC9	NOD9	NOO8	2	3.00	4.00	5.00	0.00 +	0.00	0.900
AR10	NOO6	NO10	2	4.00	5.00	6.00	0.00 +	0.00	1.000
AR11	NOO6	ND14	3	15.00	15.00	15.00	0.00 +	0.00	1.000
AR12	NOD7	NO11	2	4.00	5.00	6.00	0.00 +	0.00	0.800
AR13	NOD7	ND14	3	15.00	15.00	15.00	0.00 +	0.00	1.000
AR14	NOD8	ND12	2	4.00	5.00	6.00	0.00 +	0.00	0.600
AR15	NOD8	ND14	3	15.00	15.00	15.00	0.00 +	0.00	1.000
AR16	ND14	ND13	2	4.00	5.00	6.00	0.00 +	0.00	0.400
AR17	ND10	NO16	2	8.00	9.00	10.00	0.00 +	0.00	1.000
AR18	ND11	ND17	2	8.00	9.00	10.00	0.00 +	0.00	1.000
AR19	NO11	ND15	3	5.00	5.00	5.00	0.00 +	0.00	1.000
AR20	ND12	ND18	2	8.00	9.00	10.00	0.00 +	0.00	1.000
AR21	ND12	ND15	3	5.00	5.00	5.00	0.00 +	0.00	1.000
AR22	NO13	ND19	2	8.00	9.00	10.00	0.00 +	0.00	1.000
AR23	ND13	ND15	3	5.00	5.00	5.00	0.00 +	0.00	1.000
AR24	ND15	ND20	2	8.00	9.00	10.00	0.00 +	0.00	1.000

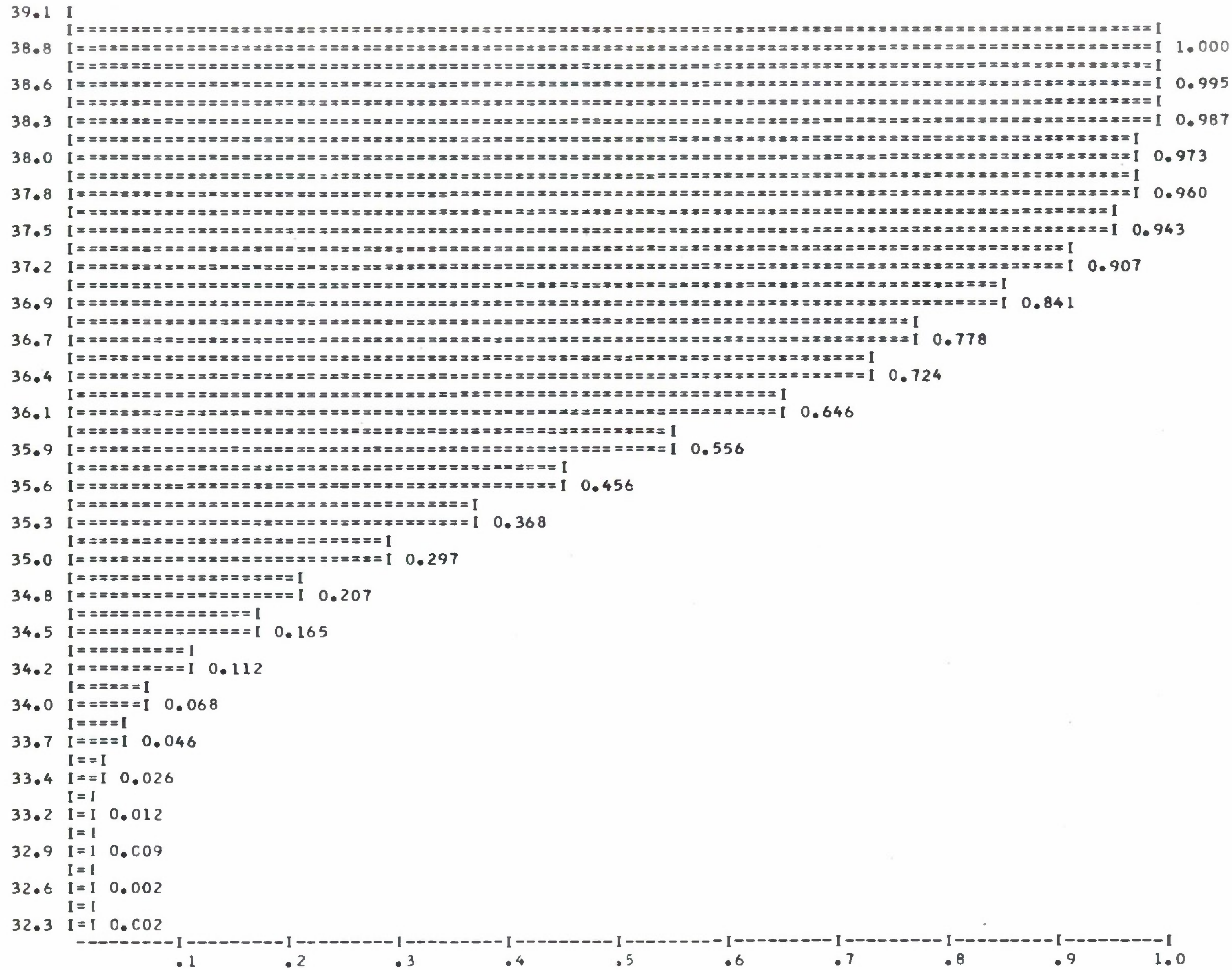
NODE	NO. OF INPUT ARCS	NO. OF OUTPUT ARCS	INPUT TYPE	OUTPUT TYPE
NOO1	0	1	4	1
NOO2	1	2	1	2
NOO3	2	2	5	5
NOO4	1	1	1	1
NOD5	2	2	6	6
NOD6	2	2	6	5
NOD7	2	2	6	6

NOD9	1	1	1	1
NOD8	2	2	6	6
ND10	1	1	1	1
ND14	3	1	2	1
ND11	2	2	6	6
ND12	2	2	6	6
ND13	2	2	6	6
ND16	1	0	1	4
ND17	1	0	1	4
ND15	3	1	2	1
ND18	1	0	1	4
ND19	1	0	1	4
ND20	1	0	1	4

```
*
 YOU HAVE RETURNED TO MONITOR MODE
 SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE
*
```

```
39.1 I
     I=I
38.8 I=I 0.004
     I=I
38.6 I=I 0.007
     I=I
38.3 I=I 0.014
     I=I
38.0 I=I 0.012
     I=I
37.8 I=I 0.017
     I==I
37.5 I==I 0.036
     I======I
37.2 I======I 0.065
     I======I
36.9 I======I 0.063
     I====I
36.7 I====I 0.053
     I======I
36.4 I======I 0.078
     I========I
36.1 I========I 0.090
     I========I
35.9 I========I 0.100
     I========I
35.6 I========I 0.087
     I======I
35.3 I======I 0.070
     I========I
35.0 I========I 0.090
     I====I
34.8 I====I 0.041
     I====I
34.5 I====I 0.053
     I====I
34.2 I====I 0.043
     I==I
34.0 I==I 0.021
     I=I
33.7 I=I 0.019
     I=I
33.4 I=I 0.014
     I=I
33.2 I=I 0.002
     I=I
32.9 I=I 0.007
     I
32.6 I
     I=I
32.3 I=I 0.002
     ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
             .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS      35.97  THE VARIANCE IS       1.37  THE MEDIAN IS      36.08  THE MODE IS      35.99
```



THE MEAN IS 35.97 THE VARIANCE IS 1.37 THE MEDIAN IS 34.96 THE MODE IS 35.99

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE N016

THE NEW DRIVING TO WORK PROBLEM

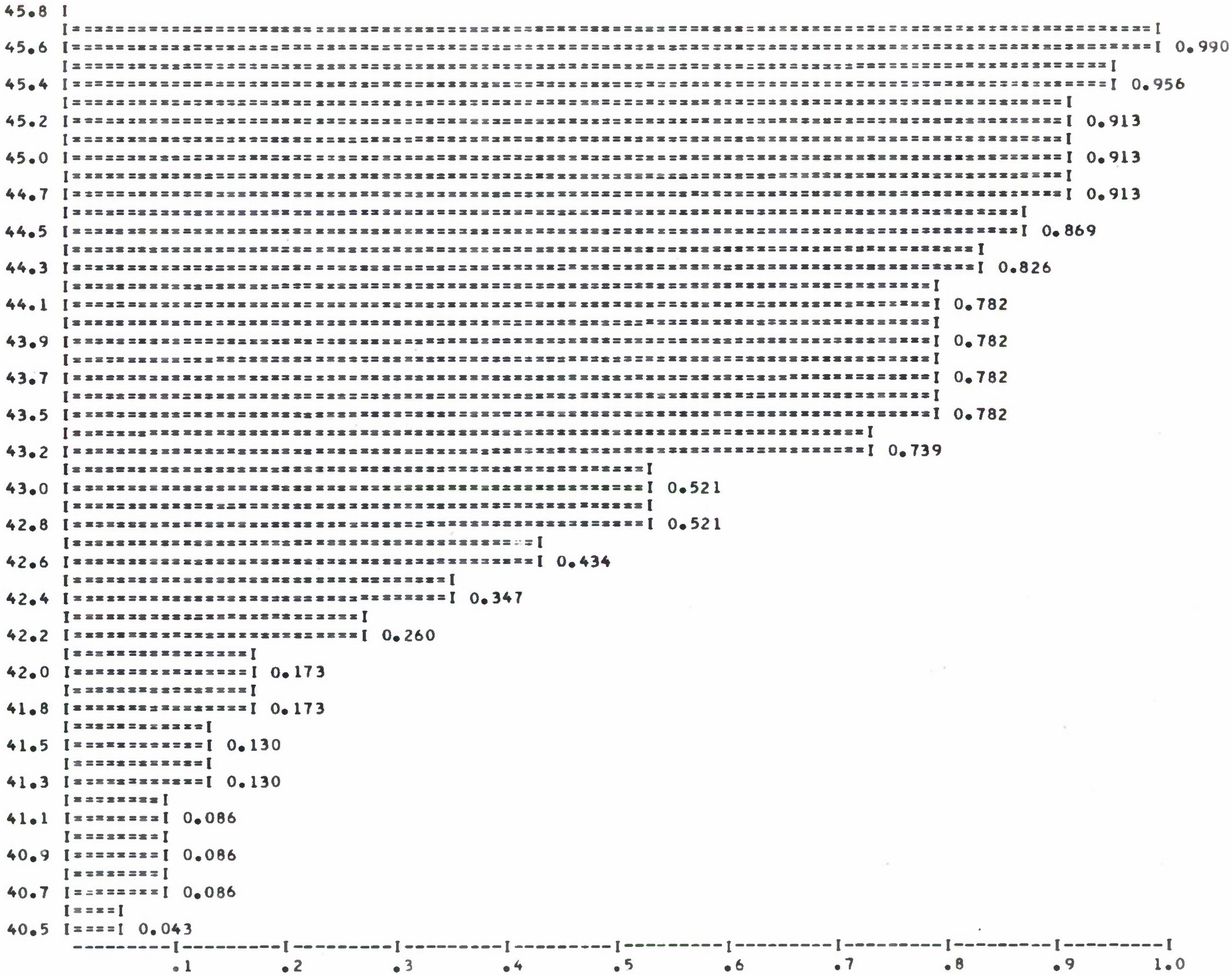
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =   0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND16

THE NEW DRIVING TO WORK PROBLEM

```
45.8 I
     I====I
45.6 I====I 0.043
     I====I
45.4 I====I 0.043
     I
45.2 I
     I
45.0 I
     I====I
44.7 I====I 0.043
     I====I
44.5 I====I 0.043
     I====I
44.3 I====I 0.043
     I
44.1 I
     I
43.9 I
     I
43.7 I
     I====I
43.5 I====I 0.043
     I====================I
43.2 I====================I 0.217
     I
43.0 I
     I========I
42.8 I========I 0.086
     I========I
42.6 I========I 0.086
     I========I
42.4 I========I 0.086
     I========I
42.2 I========I 0.086
     I
42.0 I
     I====I
41.8 I====I 0.043
     I
41.5 I
     I====I
41.3 I====I 0.043
     I
41.1 I
     I
40.9 I
     I====I
40.7 I====I 0.043
     I====I
40.5 I====I 0.043
     ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
             .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS      43.09  THE VARIANCE IS       1.72  THE MEDIAN IS      43.00  THE MODE IS      43.36
```



THE MEAN IS 43.09 THE VARIANCE IS 1.72 THE MEDIAN IS 41.70 THE MODE IS 43.36

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE NO17

THE NEW DRIVING TO WORK PROBLEM

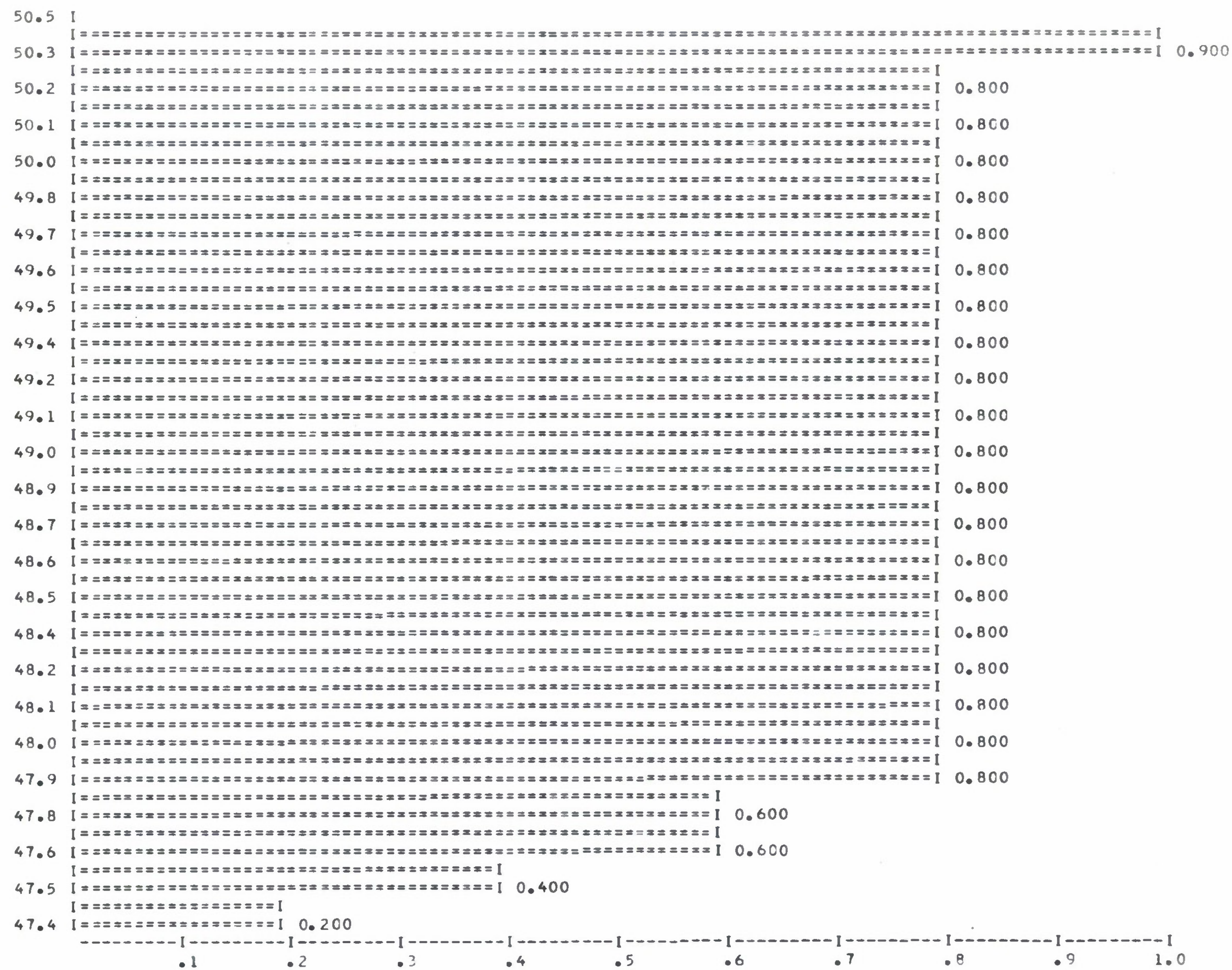
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NOOE ND17

THE NEW ORIVING TO WORK PROBLEM

```
50.5 I
     I==================I
50.3 I==================I 0.200
     I
50.2 I
     I
50.1 I
     I
50.0 I
     I
49.8 I
     I
49.7 I
     I
49.6 I
     I
49.5 I
     I
49.4 I
     I
49.2 I
     I
49.1 I
     I
49.0 I
     I
48.9 I
     I
48.7 I
     I
48.6 I
     I
48.5 I
     I
48.4 I
     I
48.2 I
     I
48.1 I
     I
48.0 I
     I==================I
47.9 I==================I 0.200
     I
47.8 I
     I==================I
47.6 I==================I 0.200
     I==================I
47.5 I==================I 0.200
     I==================I
47.4 I==================I 0.200
      ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
              .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS      48.20  THE VARIANCE IS       1.30  THE MEDIAN IS      47.71  THE MODE IS      50.40
```



THE MEAN IS 48.20 THE VARIANCE IS 1.30 THE MEDIAN IS 47.55 THE MODE IS 50.4C

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND18

THE NEW DRIVING TO WORK PROBLEM

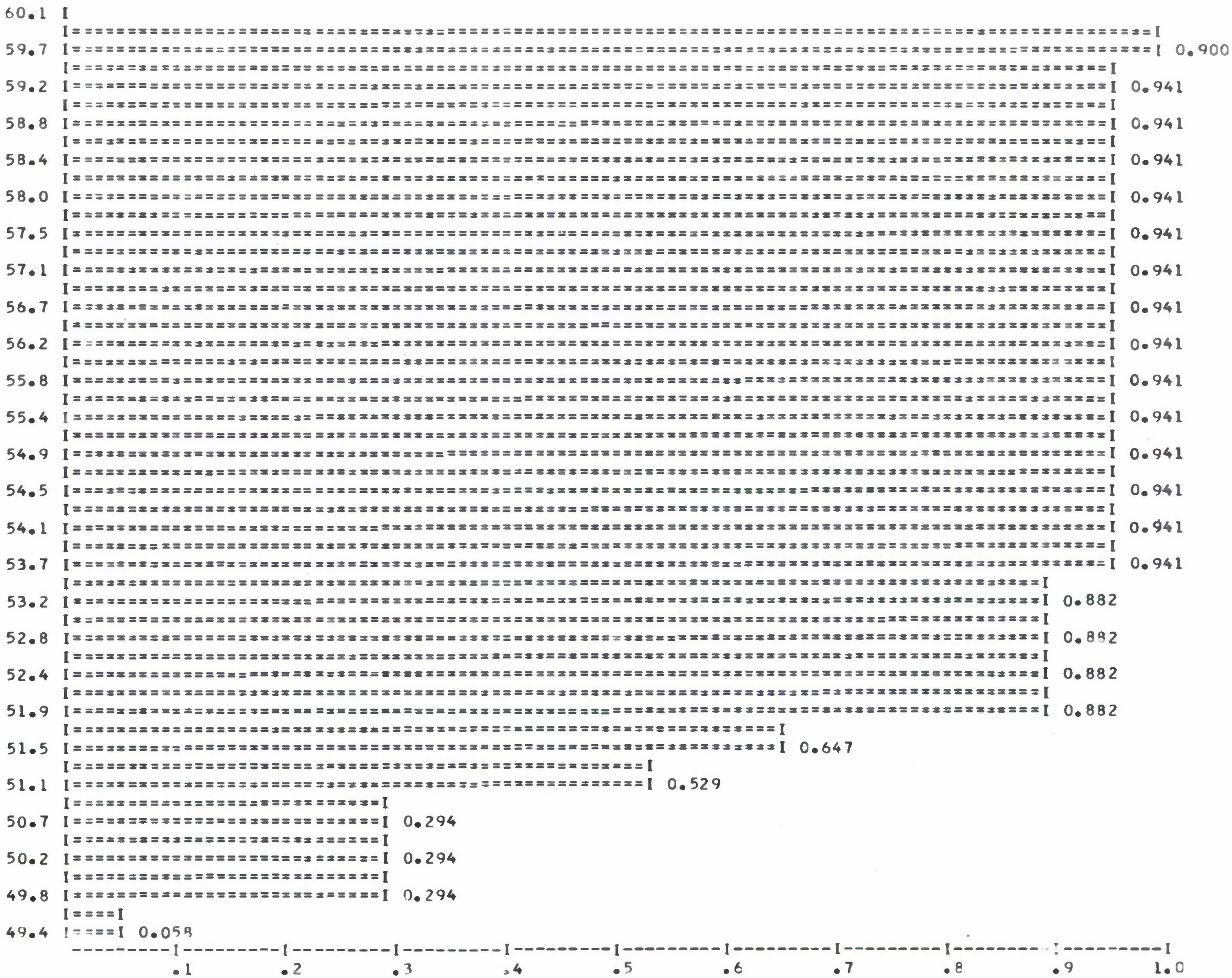
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =   0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND18

THE NEW DRIVING TO WORK PROBLEM

```
60.1 I
     I====I
59.7 I====I 0.058
     I
59.2 I
     I
58.8 I
     I
58.4 I
     I
58.0 I
     I
57.5 I
     I
57.1 I
     I
56.7 I
     I
56.2 I
     I
55.8 I
     I
55.4 I
     I
54.9 I
     I
54.5 I
     I
54.1 I
     I====I
53.7 I====I 0.058
     I
53.2 I
     I
52.8 I
     I
52.4 I
     I======================I
51.9 I======================I 0.235
     I===========I
51.5 I===========I 0.117
     I======================I
51.1 I======================I 0.235
     I
50.7 I
     I
50.2 I
     I======================I
49.8 I======================I 0.235
     I====I
49.4 I====I 0.058
      ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
              .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS      51.81  THE VARIANCE IS      5.62  THE MEDIAN IS      51.32  THE MODE IS      52.15
```



THE MEAN IS 51.81 THE VARIANCE IS 5.62 THE MEDIAN IS 50.46 THE MODE IS 52.15

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE NO19

THE NEW DRIVING TO WORK PROBLEM

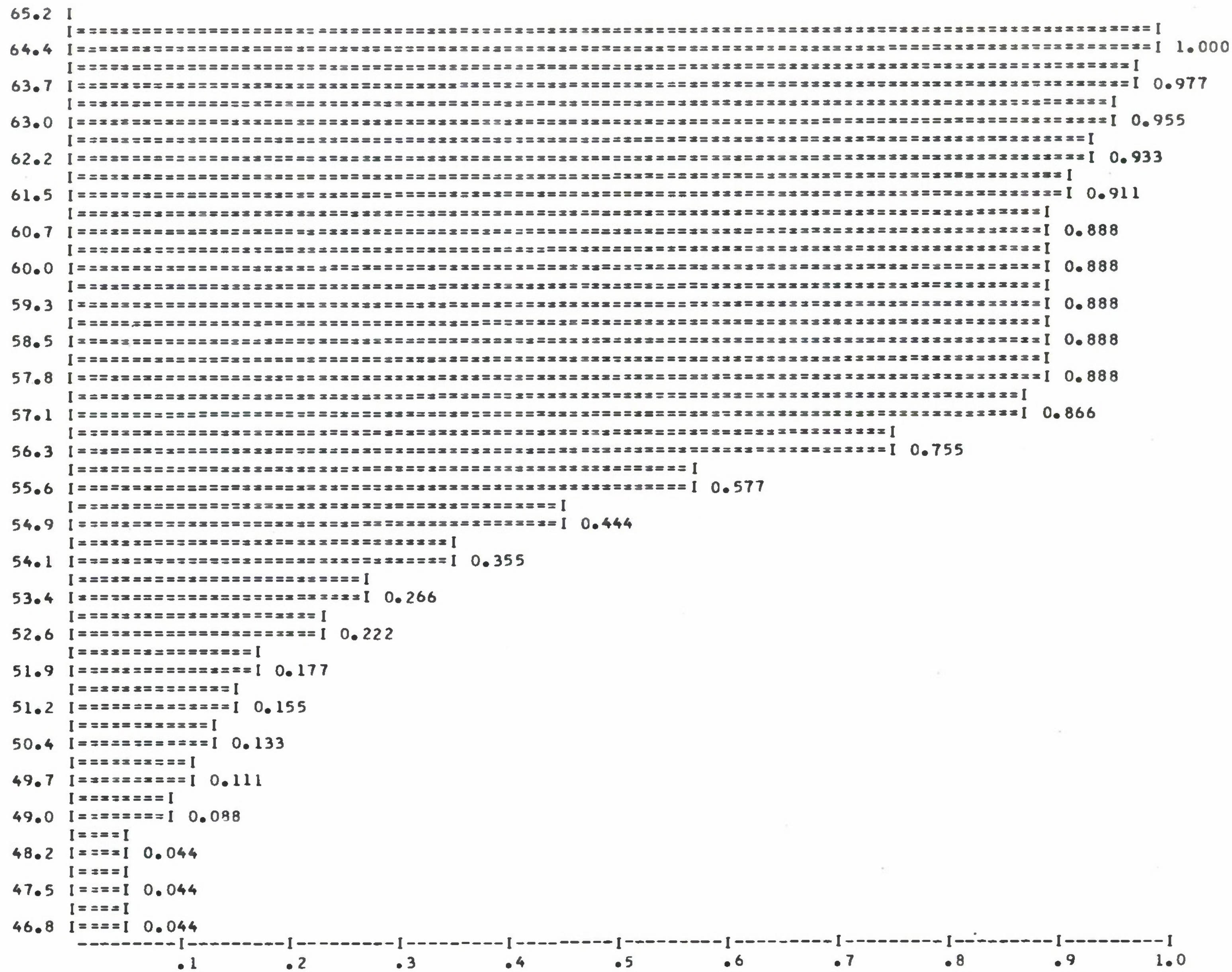
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.C000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND19

THE NEW DRIVING TO WORK PROBLEM

```
65.2 I
     I==I
64.4 I==I 0.022
     I==I
63.7 I==I 0.022
     I==I
63.0 I==I 0.022
     I==I
62.2 I==I 0.022
     I==I
61.5 I==I 0.022
     I
60.7 I
     I
60.0 I
     I
59.3 I
     I
58.5 I
     I==I
57.8 I==I 0.022
     I==========I
57.1 I==========I 0.111
     I================I
56.3 I================I 0.177
     I============I
55.6 I============I 0.133
     I========I
54.9 I========I 0.088
     I========I
54.1 I========I 0.088
     I====I
53.4 I====I 0.044
     I====I
52.6 I====I 0.044
     I==I
51.9 I==I 0.022
     I==I
51.2 I==I 0.022
     I==I
50.4 I==I 0.022
     I==I
49.7 I==I 0.022
     I====I
49.0 I====I 0.044
     I
48.2 I
     I
47.5 I
     I====I
46.8 I====I 0.044
      ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
               .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS      55.57  THE VARIANCE IS      14.97  THE MEDIAN IS      56.16  THE MODE IS      56.70
```



THE MEAN IS 55.57 THE VARIANCE IS 14.97 THE MEDIAN IS 51.72 THE MODE IS 56.70

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND20

THE NEW DRIVING TO WORK PROBLEM

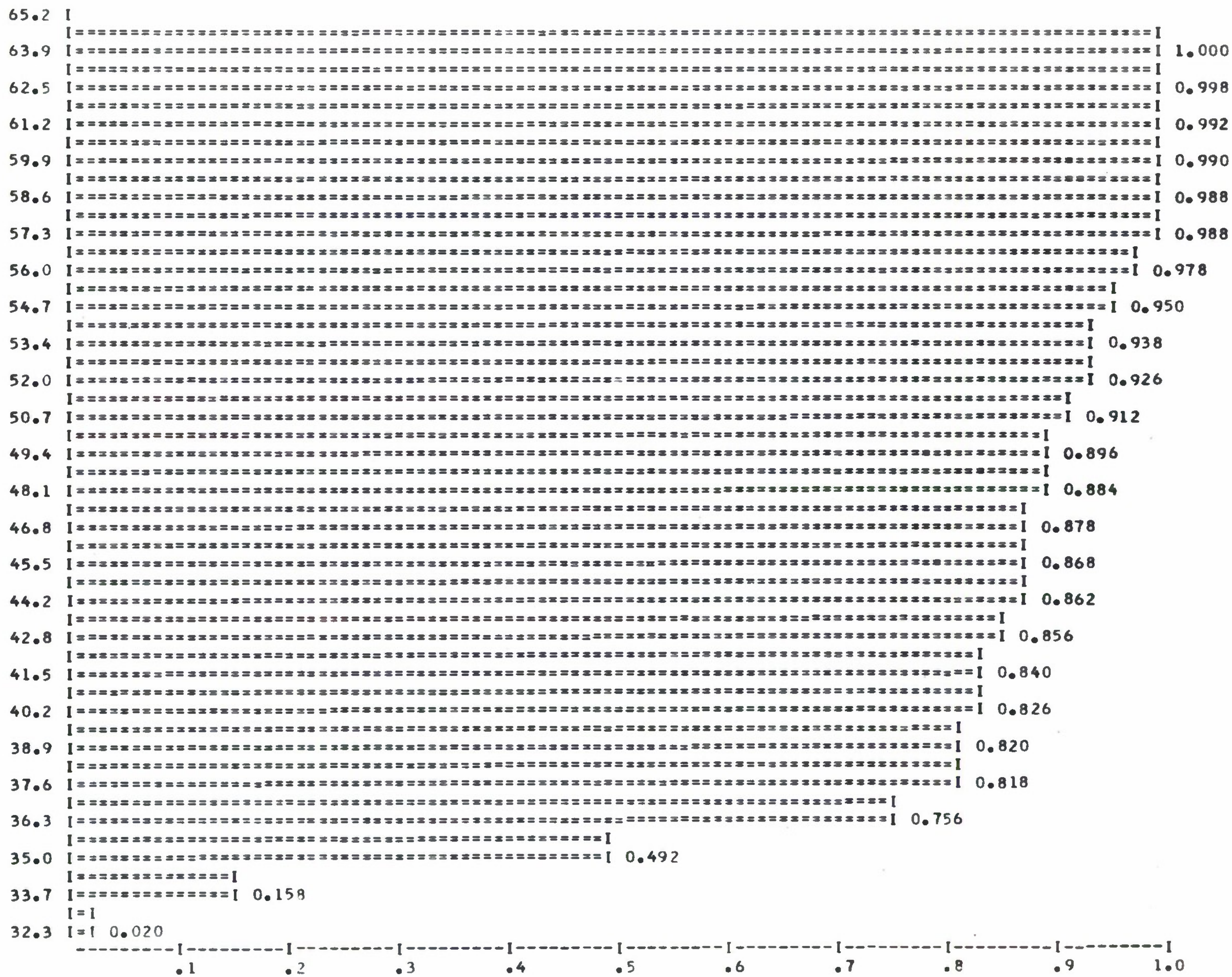
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND20

THE NEW DRIVING TO WORK PROBLEM

```
65.2 I
     I=I
63.9 I=I 0.C02
     I=I
62.5 I=I 0.C06
     I=I
61.2 I=I 0.002
     I=I
59.9 I=I 0.002
     I
58.6 I
     I=I
57.3 I=I 0.010
     I==I
56.0 I==I 0.028
     I=I
54.7 I=I 0.012
     I=I
53.4 I=I 0.012
     I=I
52.0 I=I 0.014
     I=I
50.7 I=I 0.016
     I=I
49.4 I=I 0.012
     I=I
48.1 I=I 0.006
     I=I
46.8 I=I 0.010
     I=I
45.5 I=I 0.006
     I=I
44.2 I=I 0.006
     I=I
42.8 I=I 0.016
     I=I
41.5 I=I 0.014
     I=I
40.2 I=I 0.006
     I=I
38.9 I=I 0.C02
     I======I
37.6 I======I 0.062
     I==========================I
36.3 I==========================I 0.264
     I=================================I
35.0 I=================================I 0.334
     I=============I
33.7 I=============I 0.137
     I=I
32.3 I=I 0.020
      ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
               .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

THE MEAN IS       38.72  THE VARIANCE IS       42.12  THE MECIAN IS       37.14  THE MODE IS       35.63
```



THE MEAN IS 38.72 THE VARIANCE IS 42.12 THE MEDIAN IS 35.32 THE MODE IS 35.63

GRAPH OF COMPLETION TIMES FOR ALL NODES

THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =   0.C000

GRAPH OF COMPLETION COSTS FOR ALL NODES

THE NEW DRIVING TO WORK PROBLEM

```
      I=================================================================================I
ND16  I=================================================================================I.820
      I
      I
      I====I
ND017 I====I.046
      I
      I
      I=I
ND18  I=I.010
      I
      I
      I===I
ND19  I===I.034
      I
      I
      I=========I
ND20  I=========I.090
      I
      I
       ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
                .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

               GRAPH OF NODE PROBABILITIES

                         THE NEW ORIVING TO WORK PROBLEM
```



```
*
 YOU HAVE RETURNED TO MONITOR MOOE
 SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE
*
```

Next page is blank.

APPENDIX V

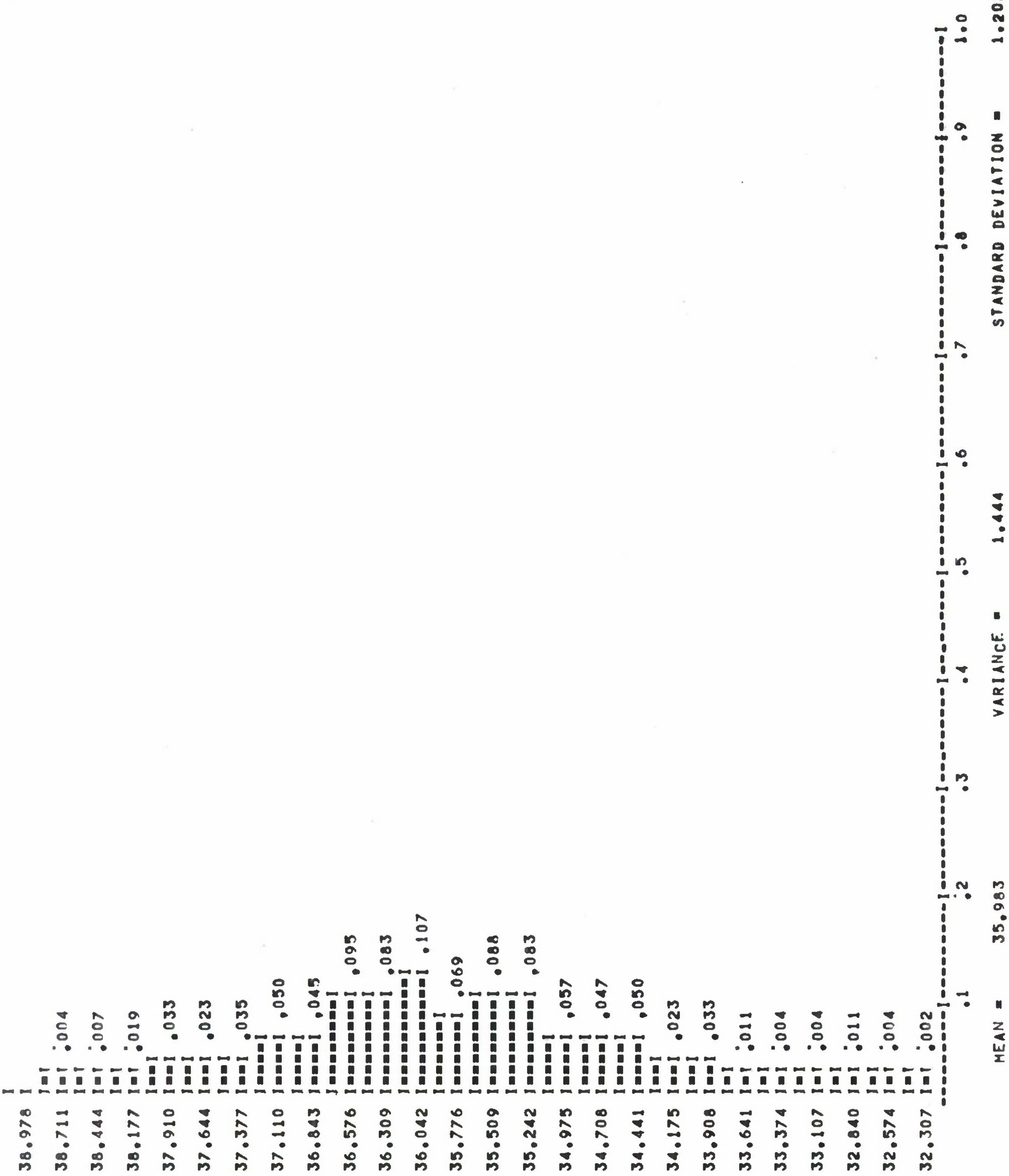
RISCA OUTPUT FOR THE GOING TO WORK PROBLEM

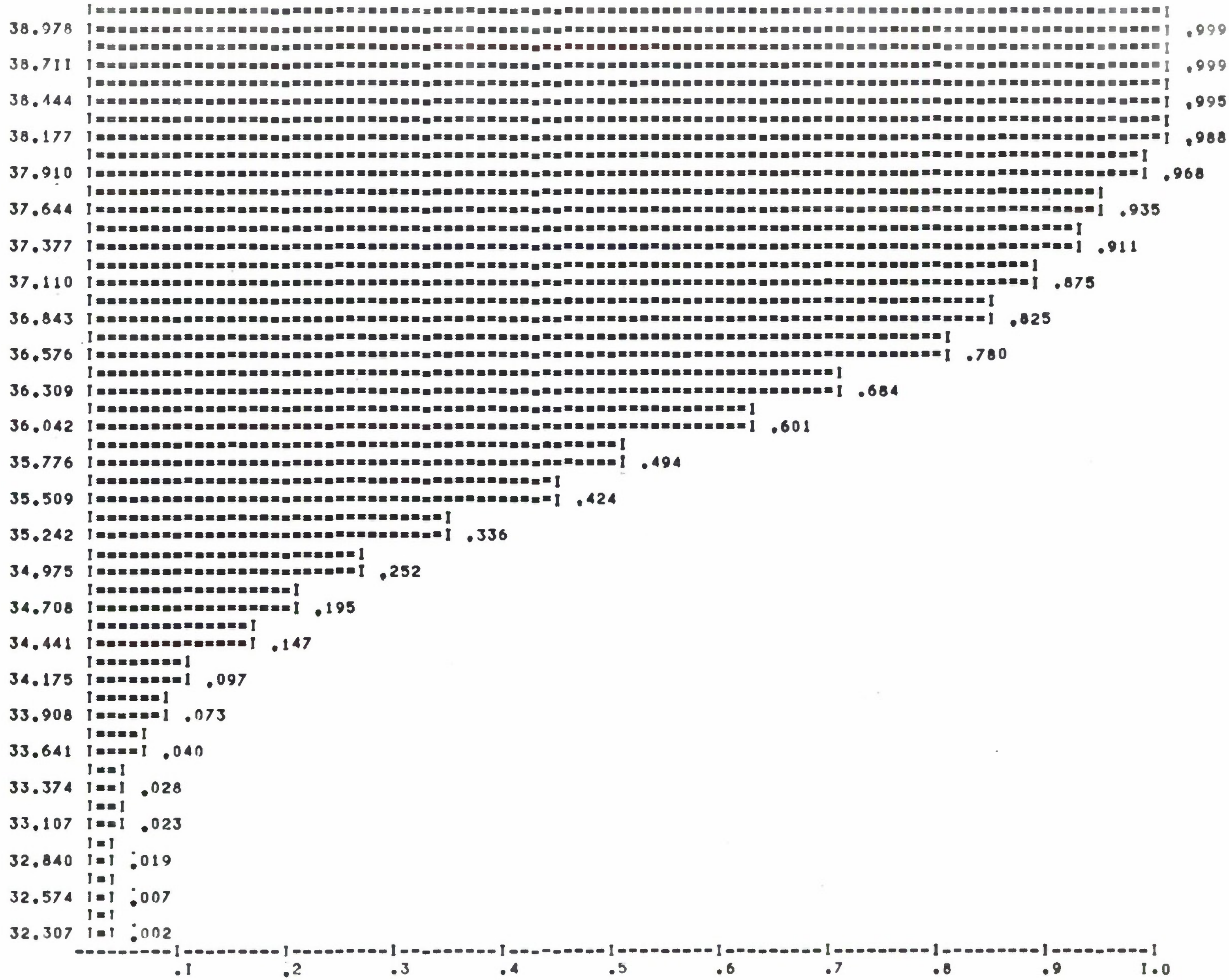
 Next page is blank.

THE NEW DRIVING TO WORK PROBLEM
500 ITERATIONS

ARC	INP NODE	OUT NODE	TIME DIST	ARG1	ARG2	ARG3	COST	P OF COMP
ARC1	NOD1	NOD2	TRI	10.00	12.00	14.00	0.00 + 0.00 T	1.00
ARC2	NOD2	NOD3	TRI	5.00	6.00	7.00	0.00 + 0.00 T	1.00
ARC3	NOD2	NOD3	TRI	11.00	13.00	15.00	0.00 + 0.00 T	1.00
ARC4	NOD3	NOD4	UNIF	0.00	0.00	0.00	0.00 + 0.00 T	1.00
ARC5	NOD3	NOD5	UNIF	0.00	0.00	0.00	0.00 + 0.00 T	.80
ARC6	NOD4	NOD6	TRI	3.00	4.00	5.00	0.00 + 0.00 T	.95
ARC7	NOD5	NOD7	TRI	3.00	4.00	5.00	0.00 + 0.00 T	.95
ARC8	NOD5	NOD9	UNIF	5.00	5.00	5.00	0.00 + 0.00 T	1.00
ARC9	NOD9	NOD8	TRI	3.00	4.00	5.00	0.00 + 0.00 T	.95
AR10	NOD6	ND10	TRI	4.00	5.00	6.00	0.00 + 0.00 T	1.00
AR11	NOD6	ND14	UNIF	15.00	15.00	15.00	0.00 + 0.00 T	1.00
AR12	NOD7	ND11	TRI	4.00	5.00	6.00	0.00 + 0.00 T	.80
AR13	NOD7	ND14	UNIF	15.00	15.00	15.00	0.00 + 0.00 T	1.00
AR14	NOD8	ND12	TRI	4.00	5.00	6.00	0.00 + 0.00 T	.60
AR15	NOD8	ND14	UNIF	15.00	15.00	15.00	0.00 + 0.00 T	1.00
AR16	ND14	ND13	TRI	4.00	5.00	6.00	0.00 + 0.00 T	.40
AR17	ND10	ND16	TRI	8.00	9.00	10.00	0.00 + 0.00 T	1.00
AR18	ND11	ND17	TRI	8.00	9.00	10.00	0.00 + 0.00 T	1.00
AR19	ND11	ND15	UNIF	5.00	5.00	5.00	0.00 + 0.00 T	1.00
AR20	ND12	ND18	TRI	8.00	9.00	10.00	0.00 + 0.00 T	1.00
AR21	ND12	ND15	UNIF	5.00	5.00	5.00	0.00 + 0.00 T	1.00
AR22	ND13	ND19	TRI	8.00	9.00	10.00	0.00 + 0.00 T	1.00
AR23	ND13	ND15	UNIF	5.00	5.00	5.00	0.00 + 0.00 T	1.00
AR24	ND15	ND20	TRI	8.00	9.00	10.00	0.00 + 0.00 T	1.00

NODE	NO. OF INPUT ARCS	NO. OF OUTPUT ARCS	INPUT RULE	OUTPUT RULE
NOD1	0	1	INIT	ALL
NOD2	1	2	AND	PROB
NOD3	2	2	1-1	1-1
NOD4	1	1	AND	ALL
NOD5	2	2	1-1B	1-1B
NOD6	2	2	1-1B	1-1B
NOD7	2	2	1-1B	1-1B
NOD9	1	1	AND	ALL
NOD8	2	2	1-1B	1-1B
ND10	1	1	AND	ALL
ND14	3	1	OR	ALL
ND11	2	2	1-1B	1-1B
ND12	2	2	1-1B	1-1B
ND13	2	2	1-1B	1-1B
ND16	1	0	AND	TERM
ND17	1	0	AND	TERM
ND15	3	1	OR	ALL
ND18	1	0	AND	TERM
ND19	1	0	AND	TERM
ND20	1	0	AND	TERM


38.978
38.711 .004
38.444 .007
38.177 .019
37.910 .033
37.644 .023
37.377 .035
37.110 .050
36.843 .045
36.576 .095
36.309 .083
36.042 .107
35.776 .069
35.509 .088
35.242 .083
34.975 .057
34.708 .047
34.441 .050
34.175 .023
33.908 .033
33.641 .011
33.374 .004
33.107 .004
32.840 .011
32.574 .004
32.307 .002
.1 .2 .3 .4 .5 .6 .7 .8 .9 1.0
MEAN = 35.983
VARIANCE = 1.444
STANDARD DEVIATION = 1.202



MEAN = 35.983 VARIANCE = 1.444 STANDARD DEVIATION = 1.202

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND16

THE NEW DRIVING TO WORK PROBLEM

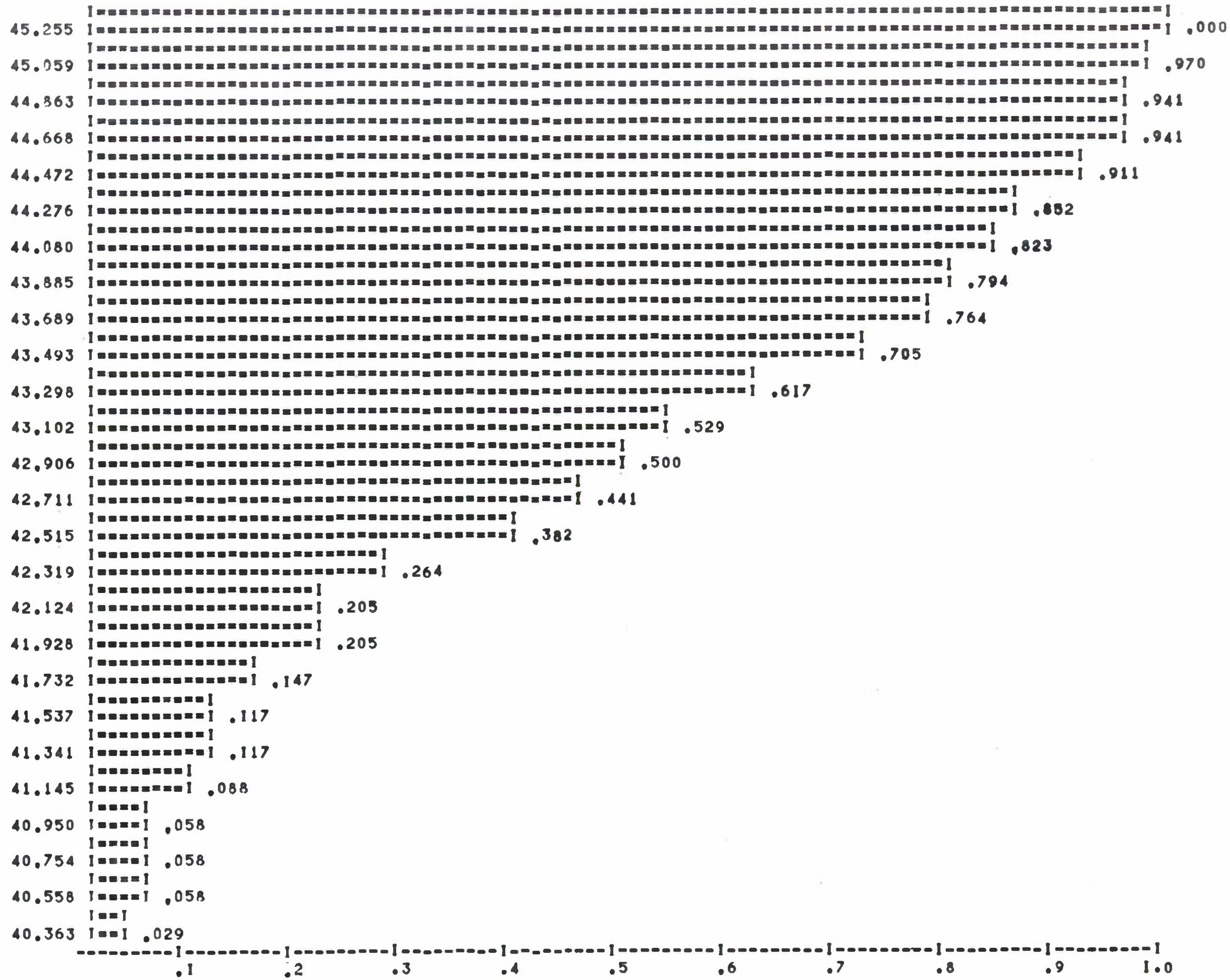
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =   0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND16

THE NEW DRIVING TO WORK PROBLEM

```
        I==I
45.255  I==I  .029
        I==I
45.059  I==I  .029
        I
44.863  I
        I==I
44.668  I==I  .029
        I====I
44.472  I====I  .058
        I==I
44.276  I==I  .029
        I==I
44.080  I==I  .029
        I==I
43.885  I==I  .029
        I====I
43.689  I====I  .058
        I=========I
43.493  I=========I  .088
        I=========I
43.298  I=========I  .088
        I==I
43.102  I==I  .029
        I====I
42.906  I====I  .058
        I====I
42.711  I====I  .058
        I===========I
42.515  I===========I  .117
        I====I
42.319  I====I  .058
        I
42.124  I
        I====I
41.928  I====I  .058
        I==I
41.732  I==I  .029
        I
41.537  I
        I==I
41.341  I==I  .029
        I==I
41.145  I==I  .029
        I
40.950  I
        I
40.754  I
        I==I
40.558  I==I  .029
        I==I
40.363  I==I  .029
        ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
                 .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

          MEAN =      43.080             VARIANCE =      1.455             STANDARD DEVIATION =      1.206
```



MEAN = 43.080 VARIANCE = 1.455 STANDARD DEVIATION = 1.206

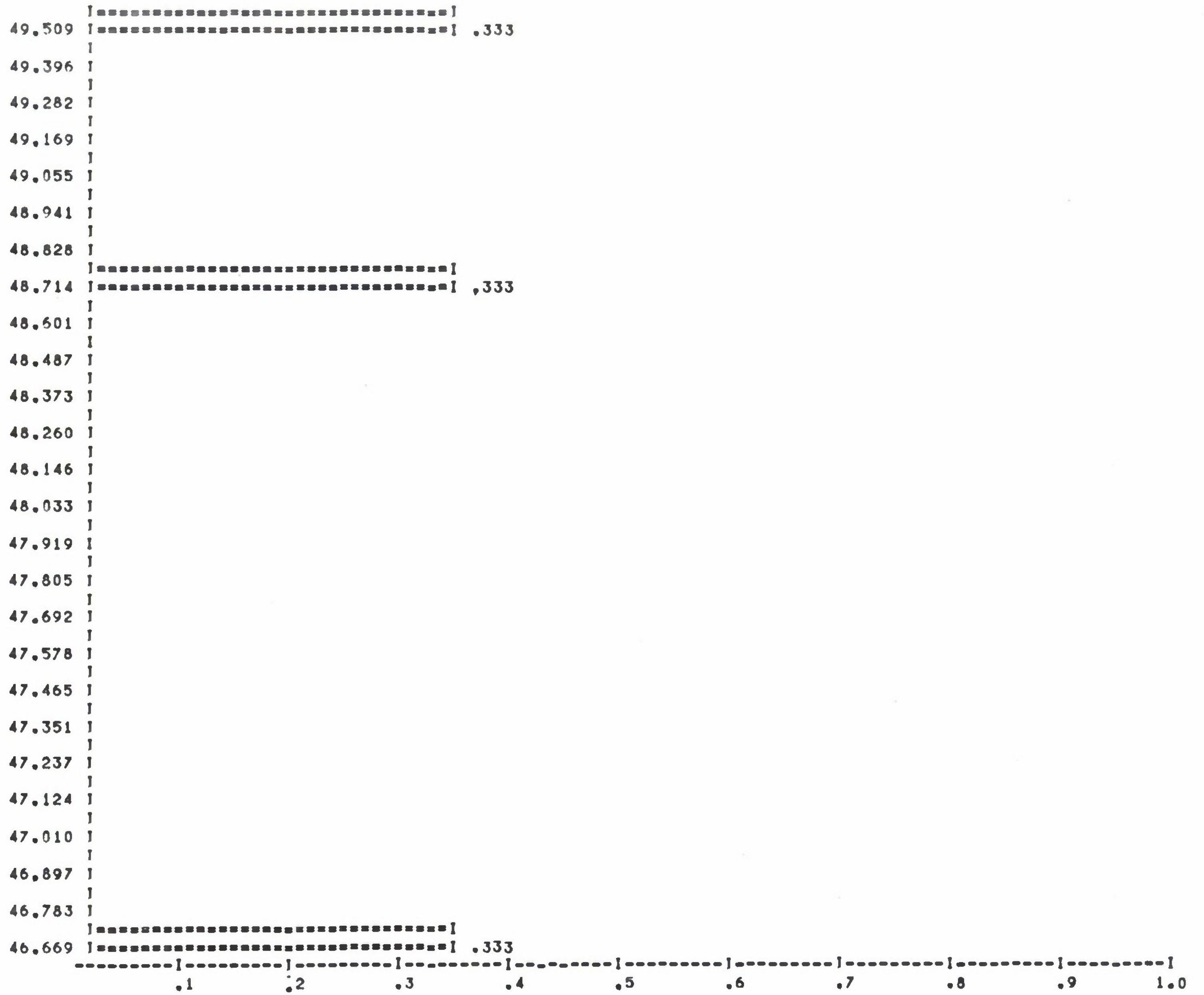
GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND17

THE NEW DRIVING TO WORK PROBLEM

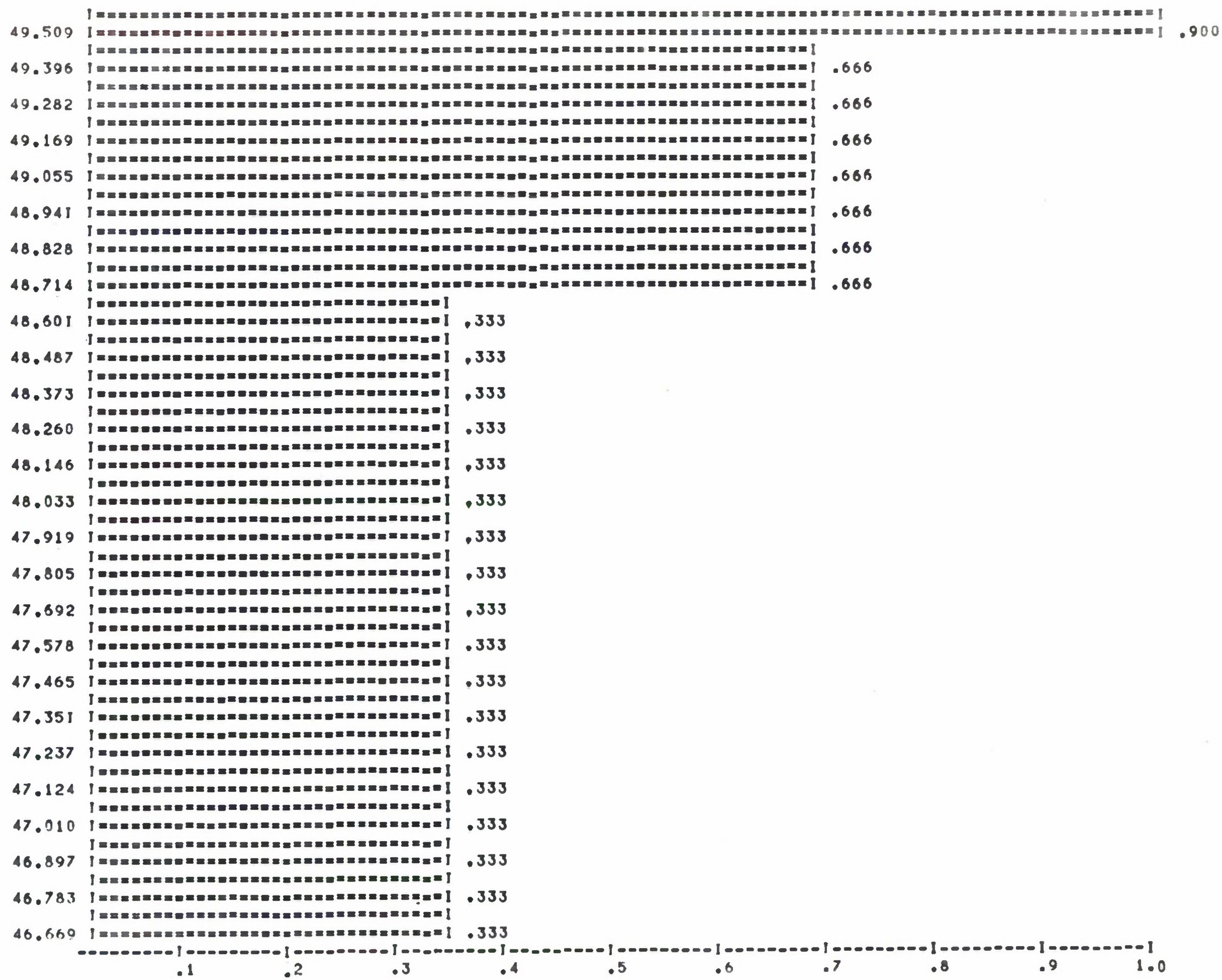
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND17

THE NEW DRIVING TO WORK PROBLEM



MEAN = 48.327 VARIANCE = 2.185 STANDARD DEVIATION = 1.478



MEAN = 48.327 VARIANCE = 2.185 STANDARD DEVIATION = 1.478

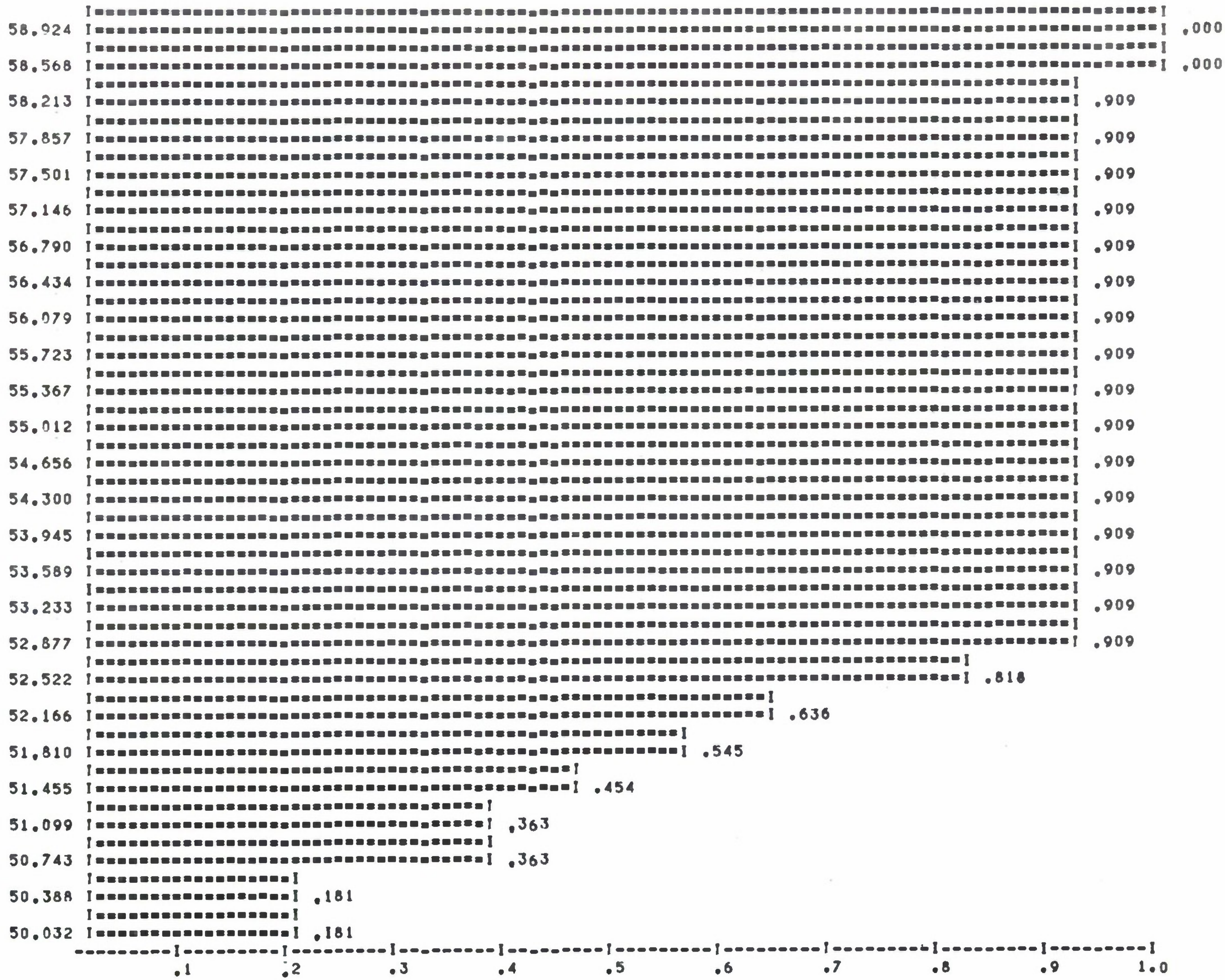
GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND18

ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND18

```
         I
58.924   I
         I=========I
58.568   I=========I .090
         I
58.213   I
         I
57.857   I
         I
57.501   I
         I
57.146   I
         I
56.790   I
         I
56.434   I
         I
56.079   I
         I
55.723   I
         I
55.367   I
         I
55.012   I
         I
54.656   I
         I
54.300   I
         I
53.945   I
         I
53.589   I
         I
53.233   I
         I=========I
52.877   I=========I .090
         I===================I
52.522   I===================I .181
         I=========I
52.166   I=========I .090
         I=========I
51.810   I=========I .090
         I=========I
51.455   I=========I .090
         I
51.099   I
         I===================I
50.743   I===================I .181
         I
50.388   I
         I===================I
50.032   I===================I .181
         ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
                 .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

           MEAN =     52.276               VARIANCE =      5.918               STANDARD DEVIATION =     2.433
```



MEAN = 52.276 VARIANCE = 5.918 STANDARD DEVIATION = 2.433

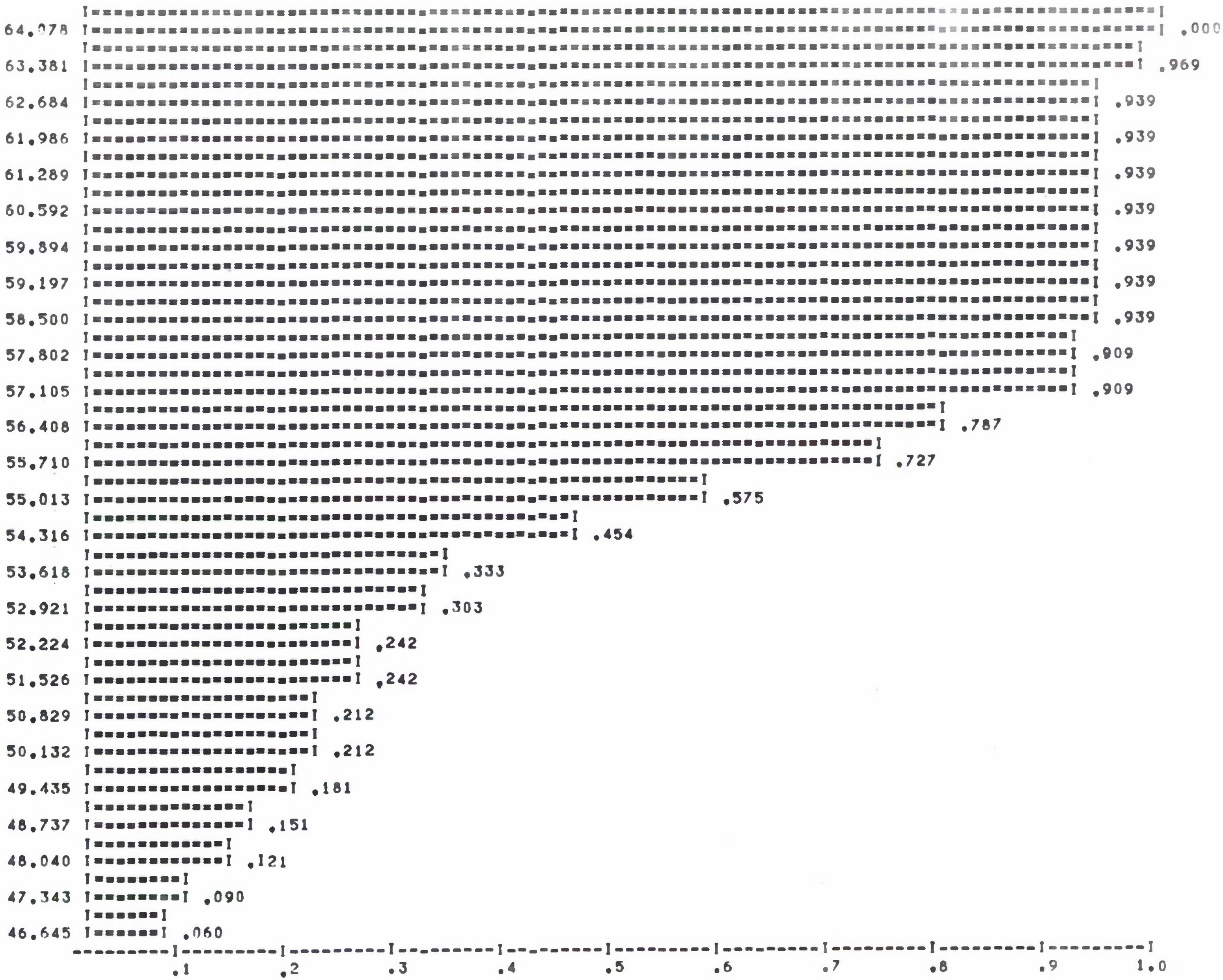
GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND19

```
                         THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =      0.0000
            GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND19
                         THE NEW DRIVING TO WORK PROBLEM
```

```
        I==I
 64.078 I==I .030
        I==I
 63.381 I==I .030
        I
 62.684 I
        I
 61.986 I
        I
 61.289 I
        I
 60.592 I
        I
 59.894 I
        I
 59.197 I
        I==I
 58.500 I==I .030
        I
 57.802 I
        I============I
 57.105 I============I .121
        I======I
 56.408 I======I .060
        I===============I
 55.710 I===============I .151
        I============I
 55.013 I============I .121
        I============I
 54.316 I============I .121
        I==I
 53.618 I==I .030
        I======I
 52.921 I======I .060
        I
 52.224 I
        I==I
 51.526 I==I .030
        I
 50.829 I
        I==I
 50.132 I==I .030
        I==I
 49.435 I==I .030
        I==I
 48.737 I==I .030
        I==I
 48.040 I==I .030
        I==I
 47.343 I==I .030
        I======I
 46.645 I======I .060
       ---------I---------I---------I---------I---------I---------I---------I---------I---------I---------I
                .1        .2        .3        .4        .5        .6        .7        .8        .9       1.0

          MEAN =      54.600              VARIANCE =      16.698              STANDARD DEVIATION =      4.086
```



MEAN = 54.600 VARIANCE = 16.698 STANDARD DEVIATION = 4.086

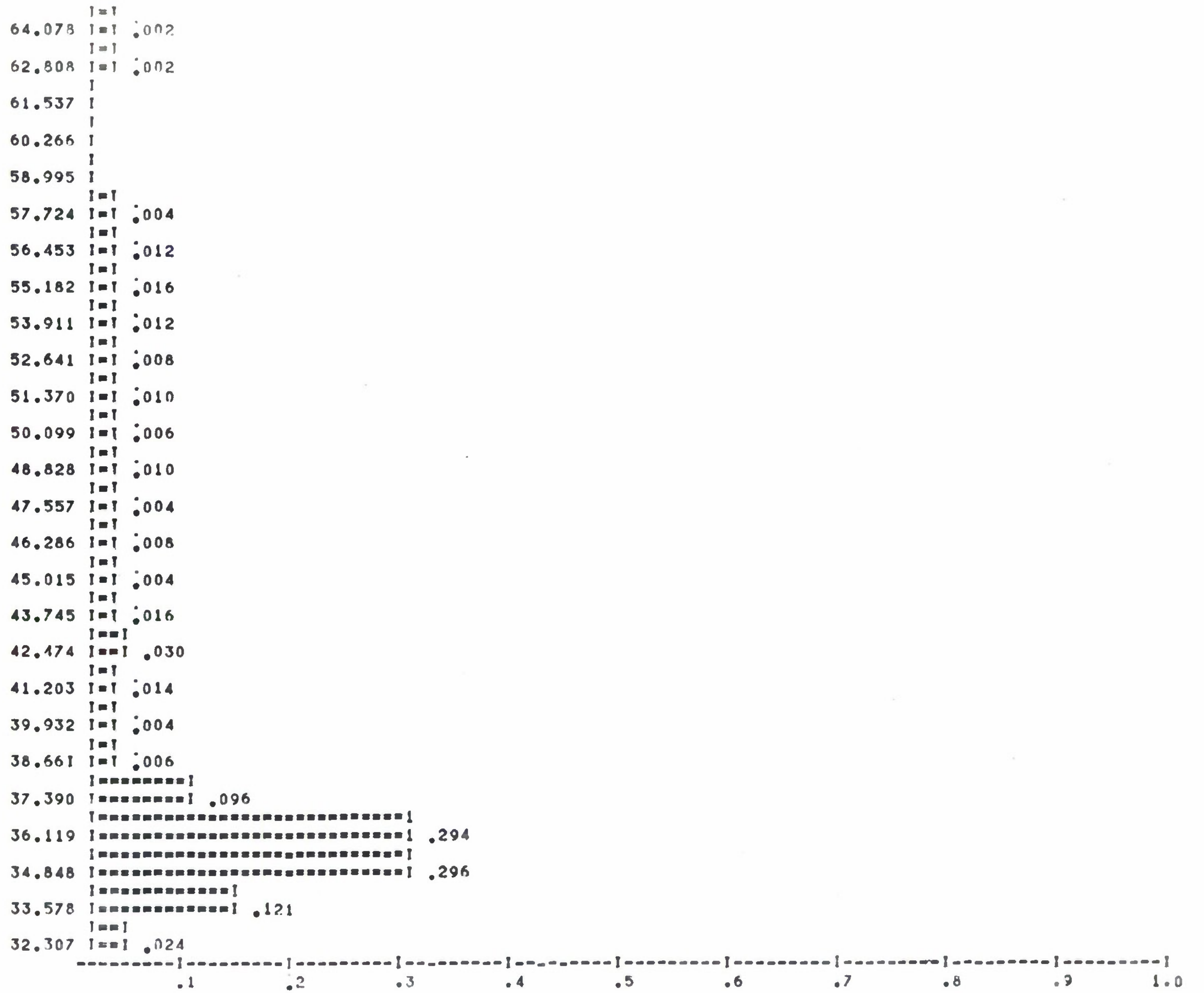
GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND20

THE NEW DRIVING TO WORK PROBLEM

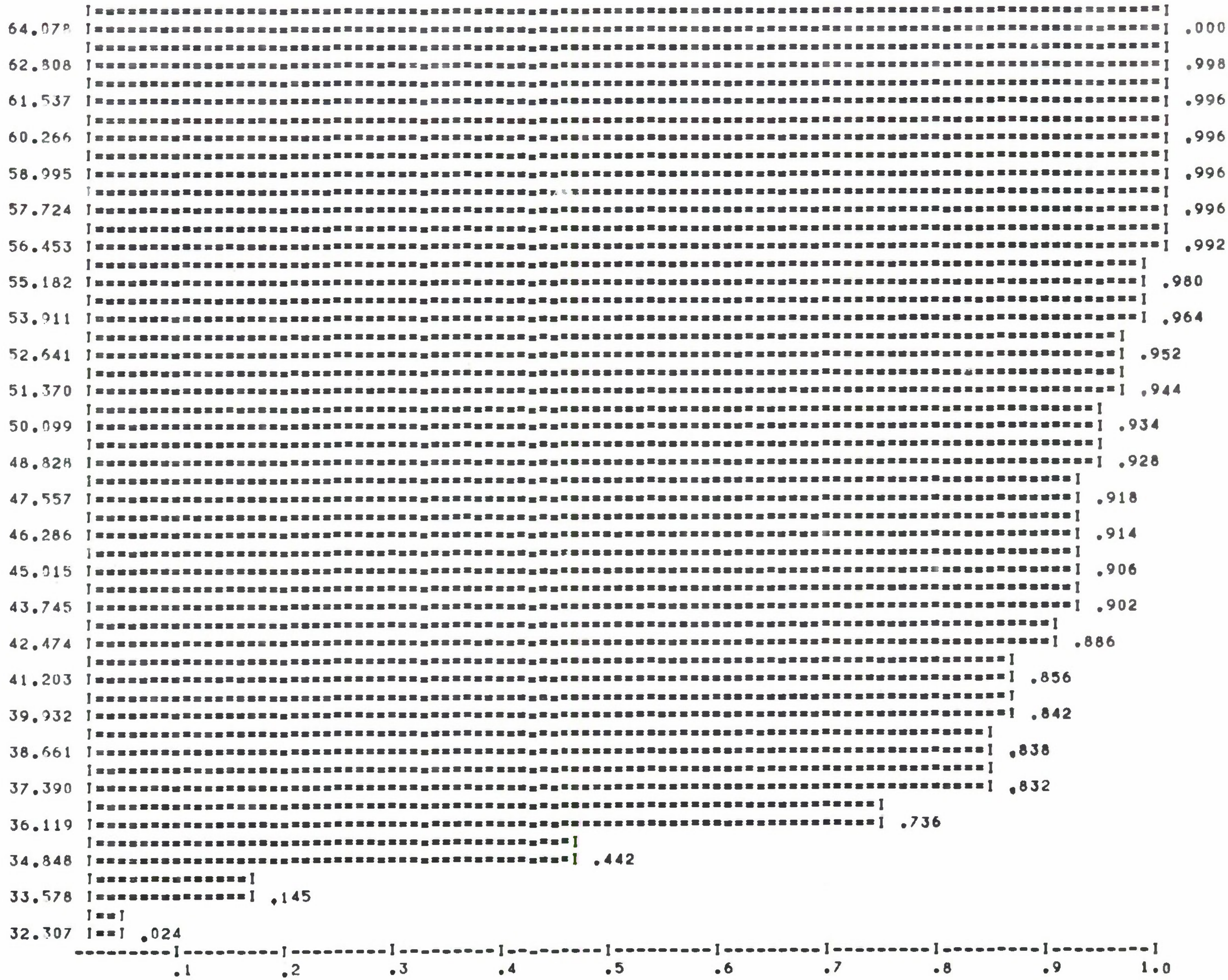
ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND20

THE NEW DRIVING TO WORK PROBLEM



MEAN = 38.127 VARIANCE = 31.020 STANDARD DEVIATION = 5.570



MEAN = 38.127 VARIANCE = 31.020 STANDARD DEVIATION = 5.570

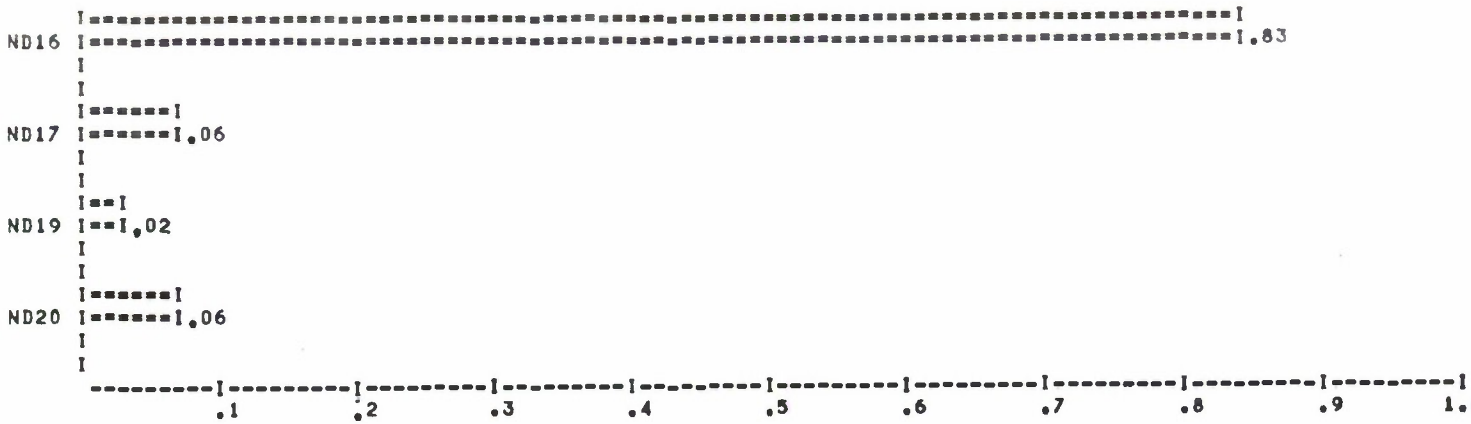
GRAPH OF COMPLETION TIMES FOR ALL NODES

THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL ,AND ARE =   0.0000

GRAPH OF COMPLETION COSTS FOR ALL NODES

THE NEW DRIVING TO WORK PROBLEM



GRAPH OF NODE PROBABILITIES

THE NEW DRIVING TO WORK PROBLEM

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: TIPCR Cameron Station Alexandria, Virginia 22314
1	Commander U.S. Army Materiel Command ATTN: AMCDL Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCMA Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCMS Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCQA Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-E Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-G Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-M Washington, D.C. 20315

No. of Copies	Organization
1	Commander U.S. Army Materiel Command ATTN: AMCRD-R Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-P Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-W Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCRD-T Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCSA-PM-MBT Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCCP Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCCP-ES (Kirk Griffin) Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCPA-SA (LTC V. Speicher) Washington, D.C. 20315
1	Commander U.S. Army Materiel Command ATTN: AMCPA-SA Picatinny Arsenal Dover, New Jersey 07801

DISTRIBUTION LIST

No. of Copies	Organization
1	Director MUCOM Operations Research Group ATTN: AMSMU-OR Edgewood Arsenal, Maryland 21010
1	Commander U.S. Army Harry Diamond Laboratories ATTN: Sys Anal Ofc Washington, D.C. 20438
1	Commander Frankford Arsenal ATTN: Cost Anal Ofc (Breen) Philadelphia, Pennsylvania 19137
1	Commander Frankford Arsenal ATTN: Sys Anal Ofc (Walters) Philadelphia, Pennsylvania 19137
1	Commander Picatinny Arsenal Plans Office ATTN: SMUPA-PA-S (J. Mydosh) Dover, New Jersey 07801
1	Chief, Analytical Sciences Office U.S. Army Biological Defense Research Laboratory ATTN: AMXBL-AS Dugway, Utah 84022
1	Commander U.S. Army Aviation Systems Command ATTN: AMSAV-R-X (Mr. Hollis) P.O. Box 209 St. Louis, Missouri 63166
1	Commander U.S. Army Electronics Command ATTN: AMSEL-PL Fort Monmouth, New Jersey 07703
1	Commander U.S. Army Missile Command ATTN: AMSMI-DA (A. Ellis) Redstone Arsenal, Alabama 35809
1	Commander U.S. Army Mobility Equipment Command ATTN: Sys Anal Ofc 4300 Goodfellow Blvd St. Louis, Missouri 63120
2	Commander U.S. Army Munitions Command ATTN: AMSMU-RE-R AMSMU-RERT (J. Selman) Dover, New Jersey
1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-CV Warren, Michigan 48090
1	Commander U.S. Army Weapons Command ATTN: Sys Anal Ofc Rock Island, Illinois 61202
1	Commander U.S. Army Management Engineering Training Agency Management Systems Division ATTN: Mr. Gerald Moeller Rock Island, Illionois 61201

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
2	Commander U.S. Army Logistics Management Center ATTN: CE&AD/SLS (Dr. Runnion) Fort Lee, Virginia 23801	1	Commander Department of Mathematics ATTN: COL Lochry U.S. Air Force Academy Colorado 80840

Aberdeen Proving Ground

Commander, USATECOM
ATTN: AMSTE-TS
AMSTE-SY (G. Schroeter)

AD MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD Brooks, Wilbert J.; Foster, Ward V. Maruyama, Richard T.

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD Network Analysis

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET. 5. RISCA.

AD Simulation

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS)
A USERS' MANUAL

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD

Brooks, Wilbert J.; Foster, Ward V.
Maruyama, Richard T.

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD

Network Analysis

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET. 5. RISCA.

AD

Simulation

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD Risk Analysis

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD MATHNET

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD RISCA

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis 2. Simulation 3. Risk Analysis 4. MATHNET 5. RISCA

AD

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNER 5. RISCA

AD Risk Analysis

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD MATHNET

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

AD RISCA

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis 2. Simulation 3. Risk Analysis 4. MATHNET 5. RISCA

AD

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED

MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&E 1P765801MM1102, November 1972

Trace Index Terms

Network Analysis
Simulation
Risk Analysis
MATHNET
RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNER 5. RISCA

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* US Army Materiel Systems Analysis Agency Aberdeen Proving Ground, Maryland 21005	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

MATHNET AND RISCA (Network Analyzer Programs) A Users' Manual

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Wilbert J. Brooks
Ward V. Foster
Richard T. Maruyama

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1972	185	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 1P765801MM1102	Technical Memorandum No. 151
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Army Materiel Command Washington, D.C.

13. ABSTRACT

Two network analyzer programs, MATHNET and RISCA, which allow the analyst to simulate a general class of network representations are described and evaluated for the potential user.

Network concepts, program listings, and program flow charts are included for both programs in addition to detailed description of input preparation and output interpretation for a hypothetical example.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
NETWORK ANALYSIS						
SIMULATION						
RISK ANALYSIS						
MATHNET						
RISCA						

Unclassified

Security Classification